

A Novel Algorithm for Text Detection and Localization in Natural Scene Images

Sezer Karaoglu, Basura Fernando
Erasmus Mundus CIMET Master
Saint Etienne, France
sezerkaraolu@yahoo.com ,basuraf@gmail.com

Alain Trémeau
Laboratoire Hubert Curien, Univ. of Saint-Etienne
Saint Etienne, France
alain.tremeau@univ-st-etienne.fr

Abstract—Text data in an image present useful information for annotation, indexing and structuring of images. The gathered information from images can be applied for devices for impaired people, navigation, tourist assistance or georeferencing business. In this paper we propose a novel algorithm for text detection and localization from outdoor/indoor images which is robust against different font size, style, uneven illumination, shadows, highlights, over exposed regions, low contrasted images, specular reflections and many distortions which makes text localization task harder. A binarization algorithm based on difference of gamma correction and morphological reconstruction is realized to extract the connected components of an image. These connected components are classified as text and non text using a Random Forest classifier. After that text regions are localized by a novel merging algorithm for further processing.

Keywords—Text Binarization; random forest classifier; Text localization; Text feature extraction

I. INTRODUCTION

In daily life, signs and texts keep important information to help making our life easier. We can emphasize the importance of texts in our life with a common instance. While we are trying to describe a place to somebody, we try to give significant clues such as the name of a street, shopping mall, cafe or business center. While trying to find the described address, we follow the signs or letters on destination which makes it efficient. Although texts keep vital role in our lives, is it always easy to make an efficient search to find out where the text stands? It is hard to define where text stands not for only automated machines but also for our brains. Underlying reason is that texts are all available on small or big areas, with a complex surrounding or simple surrounding, on the asphalt or on a zeppelin in the sky, noticeable or less noticeable but in everywhere of our live. On any artificial environment there is a large amount of textual information that we constantly use. Having this large and varying data makes the subject “text extraction from urban scenes” more complicated issue.

Being able to extract precise texts from urban scenes opens many technological development possibilities such as in mobile mapping systems to locate business in maps, self positioning in navigation systems, tourists assistants [1], or system

to help visually impaired people to move in city and perform their daily activities. Urban scenes could be analyzed simultaneously and, coupled with a text-to-speech algorithm, make them “read” the street signs, labels on shopping centers and so on. Such devices are expected soon for blind people [2][3][4].

In order to design a robust text detection algorithm, several challenges should be well understood. The main challenges can be counted as complex background, lighting, blur, resolution, occlusion, shadows, highlights, non planar objects, text size and style [7]. The main idea in this field of research is to design systems as simple as possible to be robust against these variations. Many algorithms focusing on scene text detection have been designed in the past few years. The reader may refer to [5] and [6] for a complete survey of text detection applications and systems.

Most of the previous studies in text detection can be summarized under three main categories: characters and text features, compressed and semi compressed domain, and spatial domain studies [15]. Characters and text features and compressed and semi compressed domain based studies are mostly focused on text detection through video sequences. Readers who are interested by these techniques can refer to [15]. Here this paper mainly focuses on spatial domain studies. Spatial domain studies can be classified into: edge based, texture based and connected component based. Edge based methods are focused on trying to find out regions on the image where there is a high contrast between text and background in order to detect and to merge edges from letters in images [8] [9]. Texture based methods use texture to differentiate text regions from background [10]. Basically, these methods use the knowledge that text regions in images have distinct textural properties from their backgrounds. The texture based methods mostly use texture analysis approaches such as Gaussian filtering, Wavelet-decomposition, Fourier transform, Discrete Cosine Transform (DCT), multi resolution edge detector or Gabor filtering in order to obtain texture information from images [2] [4] [11] [12]. The next step after text detection, whatever the approach used, consist to compute the energy, entropy, contrast or correlation to gather the feature of the texture. Then this information is used for

class discrimination using machine learning techniques. Some of the connected components based methods use bottom-up approach which is based on iteration to merge and combine connected pixels by the help of homogeneity criterion [14] [15]. Other methods are based either on color quantization process or on morphological operations [13]. Although, most of existing methods are efficient to detect even small size text regions in images, they suffer when images have lots of clutter especially when the text cannot be described by a homogeneity criterion. Moreover, these methods require too many heuristic rules such as aspect ratio of characters, horizontal constraint [13] [14] [15].

The problem with existing methods is that none of them is robust against many of the challenges previously mentioned. For example, while texture based methods reduce the dependency to the text size, they suffer to accurately detect boundaries of text regions. On the other hand, component based methods can quickly and accurately detect text region but they suffer when the text is embedded in a complex background or in a graphical object, they suffer also because they use too many heuristic rules.

In this paper we propose a new approach for automatically localize and extract text from indoor/outdoor images which is robust against different font size, style, uneven illumination, shadows, highlights, over exposed regions, low contrasted images, specular reflections and many distortions which makes text localization and extraction task harder. As a first step, a binarization process based on the computation of difference of gamma corrections is used to make an efficient binarization. In order to enhance image, binarization algorithm starts with suppressing the background of the image with a geodesic transform based on a connected opening morphological operator [16]. After background suppression a new image is built based on difference of gamma corrections and different gamma scales. Next, the threshold value is computed automatically by an approximation of differences of gamma from a Generalized Extreme Value Distribution [17]. Once after getting the binarized image, we set connected components (CC). Then, thanks to a random forest based learning method [24], we decide whether each connected component belongs to a text or no. As a last step of the methodology a merging step of letters is included in order to form words with letters extracted from images. The general work flow of this method is illustrated in Fig. 1.

Hereafter, the paper is structured in three parts. The section II is dedicated to a detailed description of the proposed methodology. Next, experimental results are given in section III. Lastly, a conclusion is drawn in section IV.

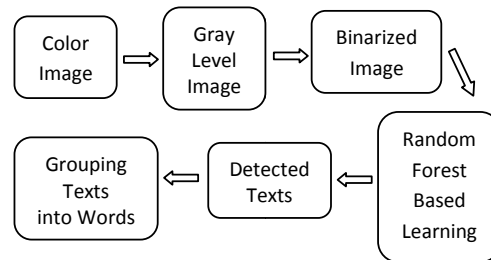


Figure 1. Flow Chart of the approach proposed

II. METHODOLOGY

The methodology proposed for text detection and extraction in indoor/outdoor scene images is supplied firstly by an binarization algorithm based on the computation of difference of gamma corrections, secondly by a random forest learning algorithm and finally letter merging (grouping). An illustrative version of the flow chart of this methodology can be seen in Fig. 2.

a.b.c.
d.e.f.



Figure 2. a. Original Input image, b. Inverted Graylevel image of "a", c. Morphological reconstructed image of "b", d. Binarization output of image "c" with proposed method, e. Machine learning process output of image "d", f. Merging output of letters which are extracted in image "e".

In the following sections, we will describe in details the main steps of the proposed methodology.

A. Image Binarization with Connected Opening and Difference of Gamma Correction

Binarization is the starting step of most image analysis system and the most crucial step of connected components (CC) based methods. If the binarization step gets irrelevant results then the whole system performance drops dramatically. It exist many methods aiming an effective binarization [18] [19] [20] [21]. Beside high-quality binarization outputs, many methods focus on low complexity and efficiency of computation of the algorithm. For this reason, we decide to use a binarization algorithm based on the computation of differences of gamma correction for image binarization [22]. In the method proposed in [22], the image is enhanced with connected opening in

order to suppress most of the background regions. After background suppression a new image is built based on difference of gamma corrections and different gamma scales. Next, a threshold value is computed from the difference of gamma images based on a statistic decision in order to get the binarized image. The threshold value is computed automatically by an approximation of differences of gamma from a Generalized Extreme Value Distribution [17]. The method proposed in [22] does not require any tuning parameter. Furthermore, it is efficiently robust different font size, style, uneven illumination, shadows, highlights, over exposed regions, low contrasted images, specular reflections, and reduces exceedingly the background noise. It is demonstrated in [22] that the proposed binarization algorithm is efficient for many cases of study as it has the desirable property of exceedingly reduce the background and noise. But, in some instances, when the natural scene images are too complex, this binarization algorithm merges nearby letters. To face this problem, we propose to use a local binarization method [23] as this latter has the desirable property to better describe the shape of connected components. In order to resolve both issues of reducing the background and noise and disconnection of nearby letters, we propose here to combine the local binarization method [23] with the global binarization algorithm proposed in [22]. The idea is to break the connectivity between letters; hence benefit from both methods.

The merging step of these two algorithms is defined as follow in (1).

$$BI(x,y)=DG(x,y)\cap LB(x,y) \quad (1)$$

Where (x,y) refers to the spatial coordinates of the current pixel, BI to the merging output, DG to the difference of gamma corrections output, and LB to the local binarization output. Fig. 3 illustrates the merging output computed from this binarization method for some of examples of text images. Once we get the BI, we refer to these connected components as text candidates. Next, we propose to use a Random forest classifier to classify these candidates as text and non text regions; this classifier is detailed in the following section.



Figure 3. From up to down, respectively, examples of text Images and binarization output associated.

B. Feature Extraction from Candidate CC's

Feature selection and parsimony is paramount in any classification task. To tackle this problem we have identified 15 independent descriptive features which characterize text from non-text candidates with significant precision and recall rates. We have explained how we obtain the text candidate CC's in previous section. In this section, we will explain how to compute these 15 independent features which will be used to classify our candidate CC's into two groups.

B. I. Geometric Features

The first category of features used is based on geometric features. They are used to measure the basic properties of CC's such as area, convex area, width, height, aspect ratio, major axis length, minor axis length, and perimeter. They are easy to calculate and helpful to quickly discard a large number of apparently non-text CC's [5].

While feature width and height refers to the width and height of the bounding box surrounding the CC's, the major and minor axis length refers respectively to the length of the major and minor axis of the ellipses that has the same normalized second central moments as the region.

Aspect Ratio feature defines the thickness of the CC's as follow in (2).

$$\text{Aspect}_{\text{Ratio}(\text{CC})} = \frac{\text{Length}(\text{minor axis}(\text{CC}))}{\text{Length}(\text{major axis}(\text{CC}))} \quad (2)$$

Feature Area counts the number of pixels in the CC's while feature Convex Area counts the number of pixels in the convex shape which completely covers the CC's. Indeed, the combination of area features with other features brings valuable information about CC's. Feature Perimeter represents the number of continuous pixels belonging to the border of the CC's. The Perimeter feature is helpful in regard to the information we want to extract considering that strokes CC's have long perimeters.

B. II. Shape Regularity Features

Next, considering that texts carry more regular shapes than arbitrary noises, we propose to use shape regularity features. We use basic shape regularity features such as occupy ratio, compactness, number of holes, solidity, roughness and filled area.

Occupy Ratio defines how much of the bounding box region is covered by CC's (3).

$$\text{Occupy}_{\text{Ratio}(\text{CC})} = \frac{\text{Area}(\text{CC})}{\text{Area}(\text{Bounding Box}(\text{CC}))} \quad (3)$$

Number Holes feature is calculated from morphological operators in order to count the number of holes in CC's.

Equiv Diameter defines the diameter of the circle having the same area as the CC's (4).

$$\text{EquivDiameter}(CC) = \sqrt{\frac{4 \times \text{Area}(CC)}{\pi}} \quad (4)$$

Compactness is a feature which divides the area of CC's by the square of CC's perimeter (5).

$$\text{Compactness}(CC) = \frac{\text{Area}(CC)}{\text{Perimeter}(CC)^2} \quad (5)$$

Filled Area defines how many percent of the CC's is not empty (6). Filled Area feature is helpful to characterize that most of texts do not have large gaps in proportion to their areas.

$$\text{Filled}_{\text{Area}(CC)} = \frac{|\text{Area}(CC) - \text{imfill}(CC)|}{\text{Area}(CC)} \quad (6)$$

Where $\text{imfill}(CC)$ is a function calculated from morphological operators which fills the holes in CC's.

B. III. Corner Based Interpolated Feature (CBIF)

We propose to use also the number of corner points. Corner points carry salient visual information and are very useful to describe some local properties of objects. Moreover, they are robust against many image deformations so they are used in many applications. We propose also to use another feature which relies on corner points. The chosen feature is a compact feature; it has the capability to capture significant shape information and to rapidly classify objects into classes. In order to detect corner points we use the methodology proposed in [29]. Once corner points are localized they are transformed to polar representation. Next, the centroid of the CC's is computed using detected corner points. Suppose that the set of corner points is given by: $C = (C_1(x,y), C_2(x,y), \dots, C_n(x,y))$ where C is the vector which keeps all corner point coordinates and (x,y) are the spatial coordinates of detected corner points. Then, the radial distance and the angle of each corner point are calculated from the centroid of corner points as follows in (7).

$$R_i = \sqrt{(y_i - y_c)^2 + (x_i - x_c)^2} \quad \theta_n = \arctan\left(\frac{y_i - y_c}{x_i - x_c}\right) \quad (7)$$

if $(x_i - x_c) > 0$ & $(y_i - y_c) > 0$

Where (x_c, y_c) are the coordinates of the centroid and (x_i, y_i) are the coordinates of the i th corner point ($i=1,2,3,\dots,n$). Then, the radial distances are normalized by (8).

$$\bar{R}_n = \frac{R_n}{\max_{i=1,\dots,n}(R_i)} \quad (8)$$

Compact shape signature is given by $f(\theta_n) = \bar{R}_n$. To construct the actual shape contour of an object, the compact shape signature is interpolated for every θ (0 to 360 degrees) using nearest neighbor interpolation. Then Fourier descriptor is applied on the interpolated shape signature to get a shape descriptor [34]. We propose to use 10 normalized Fourier coefficients. It is also possible to use other

interpolation technique. For each shape class we extract the radial signatures based on the corner points. Since very few corner points can capture shape signature; this descriptor becomes very compact and it can be used with any machine learning technique.

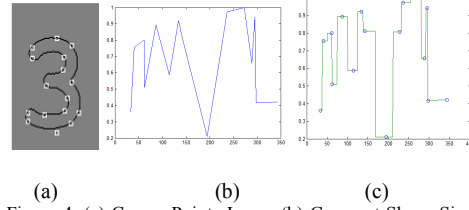


Figure 4. (a) Corner Points Image (b) Compact Shape Signature (c) Interpolated Shape Signature

The reason behind why we employ CBIF is that our experimental results show that it performs better than other classical shape descriptors when the shape is largely deformed and noise is present in the contour of the shape. Since, the binarization step deforms the contour of the shape due to various lighting conditions; this feature is used to recover the shape information with the help of interpolation.

C. Learning with Random Forest Algorithm

Several studies have proved that supervised learning based algorithms are accurate and reliable techniques well adapted to classify multi dimensional data [3] [25] [26] [27]. We propose to use the random forest classifier to classify text candidates, i.e. to decide whether candidate CC's is a letter or no, from the 15 descriptive features previously introduced. Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges to a limit as the number of trees in the forest becomes large [24]. The Random forest technique is highly accurate, handles large number of input variables and estimates the importance of variables in determining classification. Here we propose to use a Random forest classifier of 10 trees, each constructed based on 5 random features.

The classifier we used has been trained with 12672 CC's with 3031 letters and 9242 non letter instances which have been extracted from 100 images of the ICDAR training dataset [28] and labeled as letter or non-letter by hand. Since it is hard to label these CC's by hand the number of CC's for training is limited with 12672 samples.

The classical 10 fold cross validation technique has been used to evaluate the text/non-text recognition performance. An incorrectly classification rate of 4% is obtained. The results of classification step for binarized images can be seen in below Fig. 5.

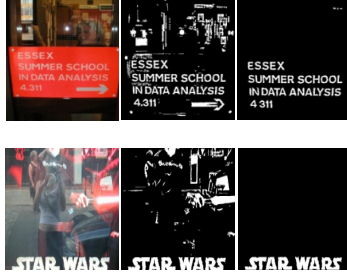


Figure 5. From left to right respectively, Original Image, Binarized Image and Detected characters after machine learning step outputs

The success of proposed learning algorithm can be seen from above images on Fig. 5. Even though the input images are complex or having various text style and size, the proposed learning algorithm can keep text regions and discard non-text regions. In some given dataset images our proposed method detects some text regions which are even not in given ground truth. When we look carefully to these images we can notice actually our algorithm does not fail. These text regions exist in images but it is not included in given ground truth. On Fig. 5 we can see one of these images. The top image on Fig. 5 includes some other text regions which is not declared in given ground truth. “Informat” texts stand on the top right of the image which is detected by our algorithm.

D. Letter Grouping in a Word

The next step of the process consists of grouping adjacent letters in order to form words. This task is one of the most difficult of text (word) extraction. In order to analyze the performance of a text extraction algorithm it is commonly recommended to compute the precision and recall rates [28]. The problem is that these performance parameters are so dependent on correctly classified words.

There are several works which try to solve this issue. While the method proposed in [31] is effective but too complicated because of training data necessity, the method proposed in [26] is simpler but not effective.

To merge adjacent letters in words we propose to use the following process which is based on the computation of distances between bounding boxes (BB) of letters detected in the previous step. The parameters used in this merging letters process are illustrated in Fig. 6. (B1, B2) represents the coordinates of the center of the two BBs of connected component.

“B1(y1a)” and “B1(y2a)” (respectively, “B2(y1a)” and “B2(y2a)”) represent the coordinates of the first BB (of the second BB) in vertical direction, “Width1” and “Width2” represent the width of the two BBs studied. “Distance” represents the distance between the centroids of the two BBs considered along the horizontal direction.

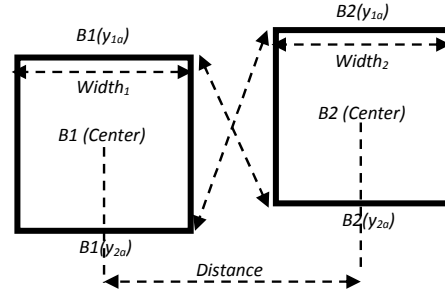


Figure 6. Parameters used in merging process.

D. I. First Step of Merging Letters

This first step of merging is based on a merging of letters along almost horizontal line. Here we have limited our study to text images whose letters are relatively well aligned, i.e. words for which the orientation of letters is aligned to within 45 degree, as for the data supplied in the ICDAR 2003 database [28].

The conditions for merging letters in detected regions are defined as below:

- $[B2(y2a) > B1(y1a)] \& [B2(y1a) < B1(y2a)]$
- $[Distance < 0.7 \times \text{Max}(Width1, Width2)]$

Any pair of BB which supplies both above conditions is then merged in this step. Two examples of merged letters are illustrated in Fig. 7.

D. II. Second Step of Merging

In the previous section, we merge the letters but did not consider if clusters of letters belong or not to the same word to separate words. The aim of this second step is to separate merged letters into words. The idea is to use a splitting criterion on previously merged CC's when there is more than one word in previously grouped CC's.

Here we propose to use a simple but effective local splitting algorithm defined as follows in (9).

$$T(i) = \text{Mean}(D(i)) + \beta \times \text{Std}(D(i)) \quad (9)$$

Threshold (T) stands for decision whether to split a group of CC's or no, while “i” stands for the block of CC's that we previously merged together. Distance (D) is a vector which keeps distances between two consecutive CC's within a block.

First, we build the distance vector by measuring the horizontal distances between CC's. Based on the statistics of distance distribution (mean and standard deviation values) over the image, a threshold is computed from (9) to split the merged connected components in two sets from distance distribution. This threshold is dependent on distances of the letters which are grouped together. If the distance between two connected components exceeds the threshold, we consider that the two CCs belongs to two different words hence they are split. The best results that we get in our experiments have

been obtained with a “ β ” value set to 1.5. Two examples of split words are illustrated in Fig. 7.

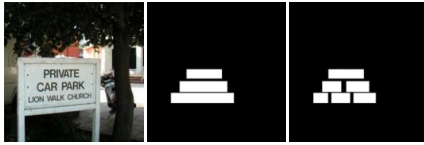


Figure 7. From left to right respectively: (a) Original image, (b) bounding boxes of merged letters after the first step, (c) bounding boxes of split words after the second step

III. EXPERIMENTAL RESULTS

A. Shape Descriptor Performance

In order to make performance evaluation of our shape descriptor “MPEG7 CE Shape-1 Part B” shape database [36] with 70 image shape classes and 1400 images are used. To represent the shape information in CBIF 10 Fourier descriptors were used. For the comparison with other algorithms we have used Centroid Based Fourier Descriptor [34] (CBFD) and Elliptic Fourier Descriptor [35] (EFD). All descriptors selected used 10 coefficients. For classifying shapes Random Forest classifier is used. Summary of experimental results on MPEG-7 shape dataset is as follows on Table 1. System precision and recall values calculated as in [37].

TABLE 1 SHAPE DESCRIPTORS PERFORMANCE

| Descriptor Performance Evaluation | Descriptors | | |
|-----------------------------------|-------------|------|------|
| | EFD | CBFD | CBIF |
| Precision | 0.54 | 0.63 | 0.8 |
| Recall | 0.64 | 0.59 | 0.78 |

B. Text Detection and Localization Performance

We perform our experiments with ICDAR 2003 challenge test dataset [28], with 249 images which contains images with various resolutions, taken both indoor and outdoor. We employ the same performance evaluation measures defined in the competition [28]. We choose this data and performance measure in order to make an objective comment on the results we obtain. The challenging and strict performance evaluation measure proposed in ICDAR metric makes researcher to use simpler images or evaluate with their own performance measures. However, ICDAR metric consider about number of detected words which makes results drastically fall down on merging steps of letters into words, widely researchers prefer to evaluate their results pixel based intersection in which they do not consider about the number of detected words. Since researchers use many different data and performance evaluation method, first we made comparison with official results presented in [28]. Our system performance is compared with the 9 other methods and shown by bars in Fig. 8.

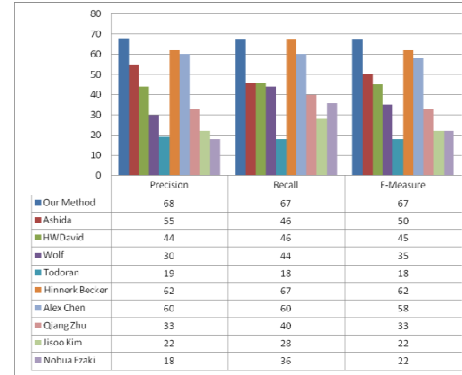


Figure 8. The performance of our algorithm with the best others reported in [28] on ICDAR 2003 Text Localization and in [30] on ICDAR 2005 Text Localization












As it can be seen from above Fig. 8, our proposed method has a significant improved results than best results presented in ICDAR 2003 [28] and in ICDAR 2005 [30] with a precision value 0.68. As we mentioned earlier, the hard dataset and strict performance measure [28] makes results drastically fall. But for the algorithm we propose in this paper main problem occurs because of strict evaluation metric of ICDAR. The proposed method works fine in almost all cases presented in ICDAR dataset. These cases can be named as shadows, degradations, non-uniform illumination, highlights, specular reflections, different font style and size and low contrast images. Even though the above results show that our method is significantly better than existing ones, in order to support our performance we evaluated our method also with pixel wise evaluation (10). “Dr” is detected text regions’ pixels and “Gr” is ground truth pixels and “Tr” is correctly detected text pixels and “Fr” is false detected text pixels and “Mr” is miss detected text pixels. “Pr” is the precision and “Rc” is the recall.

$$Pr = \frac{(Dr \cap Gr)}{(Tr + Fr)} \quad Rc = \frac{(Dr \cap Gr)}{(Tr + Mr)} \quad (10)$$

Since there is no official result on pixel wise evaluation and existing methodologies use different dataset to evaluate their system performance, it is hard to compare our results with others. The same test dataset [28] is used for pixel wise evaluation. The Precision and Recall values respectively for proposed method are calculated as 0.94 and 0.90. We propose an experiment to quantify the text extraction from natural scene images as follows. First we extract words from ICDAR2003 train images using well known commercial optical character recognition (OCR) software “ABBYY Fine Reader 10” [32]. Then output images of our algorithm are used to extract words from the same OCR software. Edit distance between the ground truth and obtained words are computed for each

image. When computing the edit distance [33]; the cost of deletion, insertion and exchange is set to 1. The experimental results based on some of the images are shown in Table 2. As it can be seen from the results our algorithm improves the text extraction of ABBYY OCR by a significant margin of 83% on the chosen dataset.

TABLE: 2 EXPERIMENTAL RESULTS FOR TEXT EXTRACTION BASED ON THE EDIT DISTANCE BETWEEN DETECTED WORDS AND THE GROUND TRUTH.IMAGE. E IS THE EDIT DISTANCE. A - USING FINE READER 10 AND B - USING FINE READER AND OUR APPROACH.

| Image | E | | Image | E | |
|---|--------------|---|---|------------|-----------|
| | A | B | | A | B |
|  | 1 | 0 |  | 6 | 0 |
|  | 3 | 0 |  | 0 | 1 |
|  | 1 | 0 |  | 27 | 0 |
|  | 17 | 0 |  | 6 | 0 |
|  | 3 | 0 |  | 11 | 6 |
|  | 41 | 0 |  | 27 | 8 |
|  | 0 | 0 |  | 0 | 0 |
|  | 1 | 1 |  | 17 | 0 |
|  | 0 | 0 |  | 30 | 16 |
| | Total | | | 191 | 32 |

IV. CONCLUSION

This paper presents a novel approach on scene text detection and localization. The method is connected component and learning based. It is robust against various conditions such as shadows, degradations, non-uniform illuminations, highlights, specular reflections, different font style

and size and low contrast images. In this method we use the utility of binarization step to discard many non-text regions while keeping text regions still in the image for more information readers should refer to [22]. We also use the utility of an effective Random Forest learning step because randomized process makes it very fast to build, it handles a very large number of input variables, it estimates the importance of variables in determining classification and it produces a highly accurate classifier [24]. However, the experimental results show that proposed method is having significant better results than existing methodologies, if the step after binarization has many text candidate CC's, it increases computation time which we aim to reduce this computation time with color information for a future work. Using color information will also help us to enhance input image when some part of the surrounding lighter than letter and some part darker because we are not robust against these cases.

As a future work we are planning to change the merging step of our algorithm with a learning based algorithm too. And we will be looking new features in order to discriminate text and non-text regions better. And finally we are planning to use larger dataset to test our method.

REFERENCES

- [1] J. Gao, J. Yang, Y. Zhang, and A. Waibel, Text Detection and Translation from Natural Scenes, Technical Report, CMU-CS-01-139, (2001)
- [2] C. M. Thillou, S. Ferreira & B. Gosselin, An embedded application for degraded text recognition, Eurasip Jour. on Applied Signal Processing, Special Issue on Advances in Intelligent Vision Systems, methods and applications, Vol. 13, pp. 2127-2135, (2005).
- [3] X. Chen and A. L. Yuille, Detecting and reading text in natural scenes. CVPR, 2:366-373, (2004).
- [4] C. M. Thillou, S. Ferreira, J. Demeyer, C. Minetti, and B. Gosselin, A multifunctional reading assistant for the visually impaired. J. Image Video Process., (4):1-11 (2007)
- [5] K. Jung, K. I. Kim, and A. K. Jain, Text information extraction in images and video: a survey, Pattern Recognition 37, no. 5, 977-997 (May 2004)
- [6] J. Liang, D. Doermann, and H. Li, Camera-based analysis of text and documents, a survey, International Journal on Document Analysis and Recognition 7, no. 2-3, 84-104 (July 2005)
- [7] C. Mancas-Thillou, Natural Scene Text Understanding. PhD thesis, Faculté Polytechnique de Mons, Belgium, (2006)
- [8] L. Agnihotri, N. Dimitrova, Text Detection for Video Analysis, IEEE Workshop on CBAIVL, pp. 109-113 (1999)
- [9] L. Agnihotri, N. Dimitrova, M. Soletic, Multi-layered Videotext Extraction Method, IEEE International Conference on Multimedia and Expo

- (ICME), Lausanne (Switzerland), August 26-29, (2002)
- [10] Z. Yu, Z. Hongjiang, A. Jain , Automatic caption localization in compressed video, *IEEE Trans. on Image Proc.*, PAMI 22, 385–392, (2000)
- [11] Ye, Q. and Q. Huang, A new text detection algorithm in images / video frames. In *Proceedings of PCM (2)*, Lecture Notes in Computer Science (LNCS), pp. 858–865, (2004)
- [12] V. Wu, R. Manmatha, and E. Riseman , Automatic text detection and recognition. In *Proceedings of Image Understanding Workshop*, 707–712, (1997)
- [13] Y.M.Y. Hasan and L.J. Karam , Morphological text extraction from images, *IEEE Trans. Image Processing* 9, no. 11, 1978-1983, (2000 November)
- [14] R. Lienhart, F. Stuber , Automatic Text Recognition in Digital Videos: *Proceedings of SPIE, Image and Video Processing IV*, Vol. 2666, pp. 180-188, (1996)
- [15] M. León, A. Gasull , Text detection in images and video sequences, *IADAT International Conference on Multi-media, Image Processing and Computer Vision*, Madrid (Spain), (March 2005)
- [16] P. Soille , *Morphological Image Analysis: Principles and Applications*, Springer, pp. 182-198, (2003)
- [17] S. Coles , *An Introduction to Statistical Modeling of Extreme Values*, Springer-Verlag editor, ISBN 1-85233-459-2, pp45-50 pp75-78, (2001)
- [18] I. Blayvas Alfred, A. Bruckstein, R. Kimmel, Efficient computation of adaptive threshold surfaces for image binarization, In *proceedings of the IEEE Computer Society Conf. On Computer Vision and Pattern Recognition*, pp 737-742, (2001)
- [19] M. Sezgin, Survey over image thresholding techniques and quantitative performance evaluation. *Journal of electronic imaging*. Vol 13(1) , pp 146-165, (2004)
- [20] P. Palumbo, P. Swaminathan, and S. Srihari , Document Image Binarization: Evaluation of Algorithms, *Proceedings of SPIE, Applications of Digital Image Processing IX*, vol. 697, pp. 278-285, (1986)
- [21] DIBCO 2009 Document Image Binarization Contest, <http://users.iit.demokritos.gr/~bgat/DIBCO2009/>
- [22] Basura Fernando, Sezer Karaoglu, Alain Tremeau , Extreme Value Theory Based Text Binarization In Documents and Natural Scenes *ICMV 2010*, in press
- [23] D. Bradley, G. Roth, Adaptive Thresholding using the Integral Image. *J. Graphics Tools* 12(2): 13-21 (2007)
- [24] L. Breiman, Random Forests. *Machine Learning*, Vol. 45(1) , pp 5–32, (2001)
- [25] B. Gatos, I. Pratikakis, K. Kepene, S.J. Perantonis , Text detection in indoor/outdoor scene images, in: *Proc. First Workshop of Camera-based Document Analysis and Recognition*, pp. 127-132, (2005)
- [26] T. Retornaz and B. Marcotegui , Scene text localization based on the ultimate opening, *International Symposium on Mathematical Morphology*, vol. 1, pp. 177–188, (2007)
- [27] J. Fabrizio, M. Cord, B. Marcotegui, Text Extraction from Street Level Images, *CMRT09 - CityModels, Roads and Traffic*. Paris, France, (2009)
- [28] ICDAR 2003 Robust reading and text locating competition image database <http://algoval.essex.ac.uk/icdar/Datasets.html>
- [29] X.C. He and N.H.C. Yung , Curvature Scale Space Corner Detector with Adaptive Threshold and Dynamic Region of Support, *Proceedings of the 17th International Conference on Pattern Recognition*, 2:791-794, (August 2004)
- [30] S. M. Lucas , Text Locating Competition Results, *ICDAR*, pp.80-85, Eighth International Conference on Document Analysis and Recognition (ICDAR'05), (2005)
- [31] Y.F. Pan, X. Hou, C.L. Liu , Text Localization in Natural Scene Images based on Conditional Random Field International Conference on Document Analysis and Recognition, (2009)
- [32] ABBYY Fine Reader 10, <http://france.abbyy.com/>
- [33] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, Vol. 10, No. 8. pp. 707-710, (1966)
- [34] D. Zhang, G. Lu , Study and evaluation of different Fourier methods for image retrieval, *Image and Vision Computing* 23 33–49, (2005).
- [35] F.P.Kuhl, C.R.Giardina , Elliptic Fourier features of a closed contour, *Computer Graphics and Image Processing*, Volume 18, Issue 3, Pages 236-258, (March 1982)
- [36] MPEG7 CE Shape-1 Part B shape database, http://www.imageprocessingplace.com/root_files_V3/image_databases.htm
- [37] Baeza-Yates, R., Ribeiro-Neto, B., *Modern Information Retrieval*. New York: ACM Press, Addison-Wesley, (1999)