Segmentation and Detection of Text in Document Images

Darko Zelenika, Janez Povh
Faculty of Information Studies, Laboratory of Data Technologies,
Ulica talcev 3, SI-8000 Novo mesto, Slovenia
{darko.zelenika, janez.povh}@fis.unm.si
Bernard Ženko
Jožef Stefan Institute, Department of Knowledge Technologies,
Jamova cesta 39, SI-1000 Ljubljana, Slovenia
bernard.zenko@ijs.si

Abstract: Text detection in document images plays an important role in optical character recognition systems and is a challenging task. The proposed text detection method uses self-adjusting bottom-up segmentation algorithm to segment a document image into a set of connected components. The segmented connected components are then described in terms of 27 features and a machine learning algorithm is used to classify these components as text or non-text. We have collected a dataset (called ASTRoID), which contains 500 images of text blocks and 500 images of non-text blocks in order to test the method. We empirically compare performance of the proposed text detection method with seven different machine learning algorithms; the best performance is obtained with the radial support vector machine.

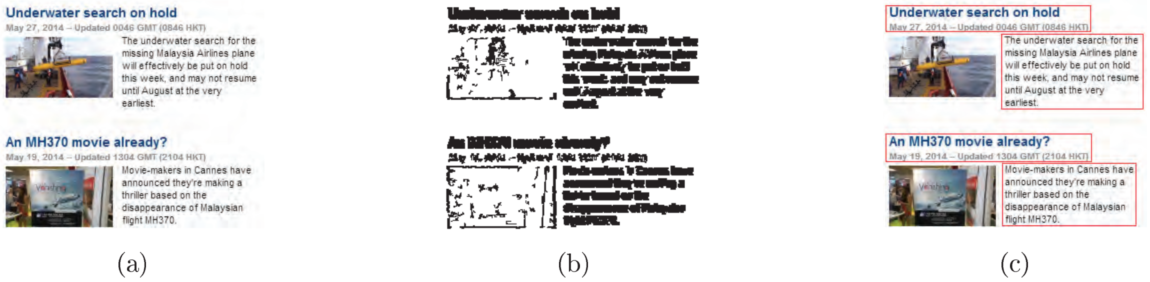
Keywords: text detection, document segmentation, text/non-text classification, machine learning

1 INTRODUCTION

Today a lot of potentially useful textual information is stored in an unstructured form of images of documents such as invoices, contracts, web pages, etc. In order to effectively recognize and extract this text with Optical Character Recognition (OCR) technology, location of the text within the image must be detected first. The first step of text detection in document images is the document segmentation, which is followed by a classification of segments obtained in the first step. Document segmentation is a task which splits a document image into segments or blocks of interest, as shown in Fig. 1b. Here, each group of black pixels (called connected component (CC)) represents one block. Blocks of interest can usually be classified as text or non-text. We are mainly interested in text blocks, so our goal is to identify them and separate them from non-text blocks (see Fig. 1c). It is important to understand that document segmentation and classification (identification) of segmented blocks can hardly be treated as independent tasks and are often merged together in a "(physical) layout analysis" [8].

Document segmentation techniques are traditionally classified into three categories: top-down, bottom-up and hybrid approaches. The top-down approach starts the segmentation from bigger blocks and repetitively segments the document image into smaller blocks until the document image is segmented into the smallest possible blocks [2, 7, 9, 12, 13, 15, 18]. The bottom-up approach is the opposite of the top-down approach. It starts from the smallest segments (characters) and then joins them into bigger and bigger blocks (words, paragraphs, etc.)[3, 4, 5, 16, 20]. An approach that does not fit into a top-down or bottom-up strategy or uses the combination of both is called a hybrid approach [10, 14]. Hybrid approaches often try to combine the speed of top-down approaches and the robustness of bottom-up approaches.

The task of the classification algorithm is to classify the results of the segmentation algorithm. The accuracy of the classification highly depends on the quality of the results of the segmentation algorithm. A standard approach for detecting text and non-text blocks is to extract features from segmented CCs in document images and classify them with some machine learning (ML) algorithm [2, 3, 5, 20]. In this paper we use a similar approach.



Underwater search on hold

May 27, 2014 -- Updated 0046 GMT (0846 HKT)

The underwater search for the missing Malaysia Airlines plane will effectively be put on hold this week, and may not resume until August at the very earliest.

An MH370 movie already?

May 19, 2014 -- Updated 1304 GMT (2104 HKT)

Movie-makers in Cannes have announced they're making a thriller based on the disappearance of Malaysian flight MH370.

(a) (b) (c)

Figure 1: a) Image document, b) Segmented image and c) Detected text

The main contributions of this paper are: (1) a self-adjusting segmentation algorithm for finding text CCs that is independent of the image resolution and font size, (2) a new set of features which describe differences between text and non-text image blocks, (3) a custom benchmark dataset ASTRoID of text and non-text image blocks, and (4) investigation of performance of seven different ML algorithms for separation between text and non-text image blocks.

The rest of the paper is organized as follows. In section two, we introduce the document segmentation algorithm, the ASTRoID dataset and the classification algorithm. The classification results obtained with different ML algorithms are presented and discussed in section three. Finally, section four concludes the paper.

2 TEXT DETECTION

Our text detection method performs two tasks: document image segmentation and classification. The segmentation algorithm extracts image blocks, these are described in terms of the selected features, and then classified with a ML algorithm as either text or non-text blocks.

2.1 Segmentation

The segmentation algorithm follows the bottom up strategy. It segments the document into CCs, which are constructed with a combination of the Sobel edge detection and dilation methods [8, 17], and is composed of three parts: (1) search for an optimal rectangular kernel, (2) edge detection and (3) extraction of standalone document image blocks. The input is a grayscale image (Fig. 2a) and the output is a binary image.

The first part of the segmentation algorithm finds an optimal rectangular kernel, which is then used by the other two parts of the algorithm. The optimal rectangular kernel highly depends on the height of the dominant text in the document image. In the second part of the segmentation algorithm we apply different Sobel kernels (vertical, horizontal and diagonal) on the grayscale document image. The resulting images of different Sobel kernels are dilated by the optimal rectangular kernel and combined into one image by logical *AND* and *OR* operations, as shown in Fig. 3a. In the third part of the segmentation algorithm we localize all CCs as illustrated in Fig. 2b (red rectangles in Fig. 2c) and keep only those CCs that intersect with two or less than two other CCs (due to characters such as "B" and "8") and we call them standalone CCs. In such a way most of CCs which do not belong to text blocks are removed. This image then is dilated by the optimal rectangular kernel, as shown in Fig. 3b. The final segmented image is obtained by combining the resulting images of the second (Fig. 3a) and third (Fig. 3b) part of the algorithm with logical *OR* operation (see Fig. 3c). In such a way we segment the document image into blocks that can either be text or non-text blocks.

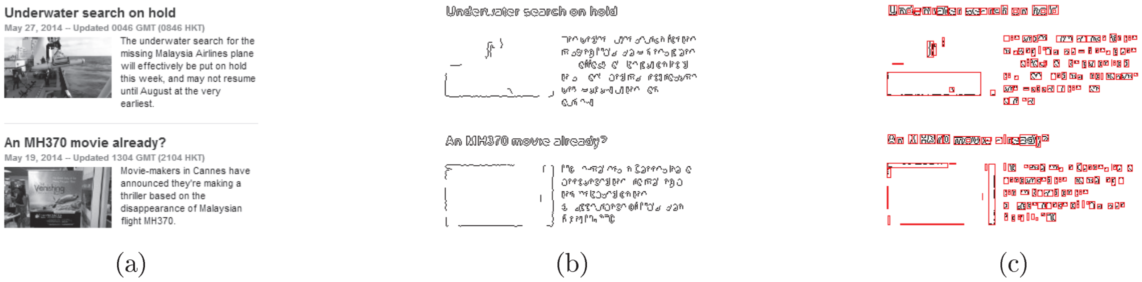

Underwater search on hold
May 27, 2014 -- Updated 0046 GMT (0846 HKT)
The underwater search for the missing Malaysia Airlines plane will effectively be put on hold this week, and may not resume until August at the very earliest.
An MH370 movie already?
May 19, 2014 -- Updated 1304 GMT (2104 HKT)
Movie-makers in Cannes have announced they're making a thriller based on the disappearance of Malaysian flight MH370.
(a)
(b)
(c)

Figure 2: a) Grayscale image, b) Sobel edges and c) Localized CCs

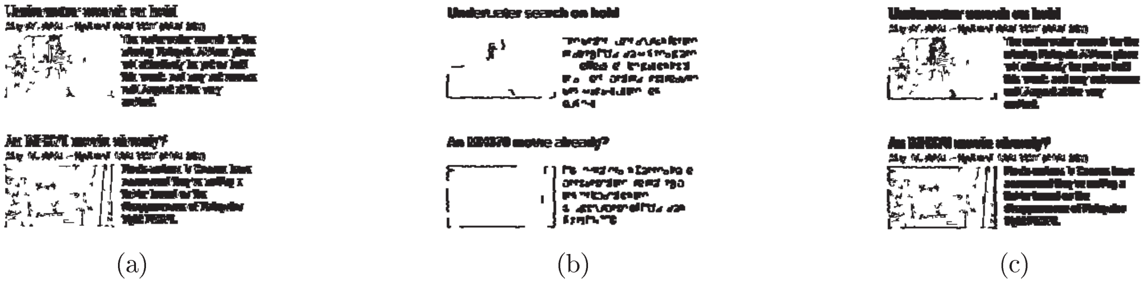


Figure 3: Segmentation steps, a) 2nd part, b) 3rd part and c) Final part

2.2 Text vs. non-text classification

In order to evaluate our method we created our custom dataset of text and non-text image blocks, which we called ASTRoID. The dataset includes 500 image blocks which contain plain text of different sizes, lengths, colors, font styles, etc., and 500 equally diverse image blocks which do not contain text. The ASTRoID dataset is available for download at: `http://dk.fis.unm.si/ASTRoID.zip`.

In document images most of the text blocks are uniformly structured and have a regular shape. On the other hand, non-text blocks have a lot of shape variability, i.e., mostly they have an irregular shape. But only shape information is not enough for classification, we also need to take into account the information on the context of these blocks. In this paper we used features similar to the ones proposed in [2, 3, 5, 12]. However, we took a different approach to calculate some of the proposed features and introduced a new feature "color density". In our approach we extract features from multiple images of the same segmented block (Fig. 4), which are obtained by different preprocessing methods, unlike the approaches found in the literature where features are extracted from usually one binary image. Before the actual feature extraction, each segmented block is resized to 100 pixels in height while maintaining the width to height aspect ratio. In our approach we used the following features: number of CCs [2], aspect ratio [2, 3, 12], foreground density [2, 3, 5, 12], color density, standard deviation of the heights and widths of CCs [3, 5, 12], and standard deviation of the lengths of horizontal and vertical runs [12]. Most of the features are extracted from different binary images (Fig. 4b - 4h), which are obtained with the following methods: skeletonization (Fig. 4d) [1], horizontal Sobel kernel (Fig. 4e), vertical Sobel kernel (Fig. 4f), diagonal Sobel kernels (Fig. 4g) and Canny edge detection method (Fig. 4h) [6, 17].

The number of CCs is computed after the binarization of grayscale image with the Otsu's [11] thresholding algorithm (Fig. 4b). The feature foreground density is calculated two times, i.e., once for Fig. 4b and once for Fig. 4h. The color density feature is calculated from the color image (Fig. 4a). The standard deviation of the heights of CCs is calculated eight times, i.e., once for each of the figures: Fig. 4b, 4c, 4d, 4f, 4g and 4h, and for the Sobel vertical lines which are extracted from Fig. 4c and 4h. The standard deviation of the widths of CCs is caluclated

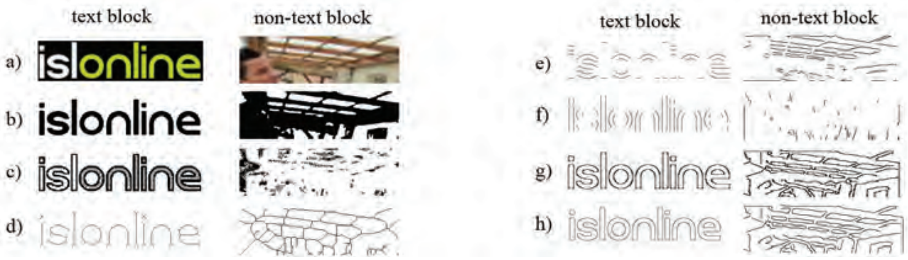


Figure 4: a) Color image, b) Binary image, c) Image of extracted colors, d) Skeleton image, e) Horizontal Sobel, f) Vertical Sobel, g) Diagonal Sobel and h) Canny

two times, i.e., once for Fig. 4e and for the Sobel horizontal lines which are extracted from Fig. 4h. Standard deviation of the vertical runs is calculated nine times, i.e., once for each of the following figures: Fig. 4b, 4c, 4d, 4f, 4g and 4h, and for the Sobel vertical lines which are extracted from Fig. 4c, 4g and 4h. Standard deviation of the horizontal runs is calculated three times, i.e., once for each of the following figures: Fig. 4b and 4e, and for the Sobel horizontal lines which are extracted from Fig. 4h.

All together we extracted 27 features from each image block of the ASTRoID dataset and assigned a class label (text or non-text) to each one. In this way we created a dataset, which we used for classification with seven popular ML algorithms that are frequently used in practical applications and typically give good results, in order to evaluate our choice of features. We used: Naïve Bayes, C4.5 (decision tree), k-Nearest Neighbors (k-NN), Random Forest, Linear Support Vector Machine (SVM), Polynomial SVM and Radial SVM. The accuracy of each of the above mentioned algorithms is estimated by 10-fold cross-validation. We used the implementations of ML algorithms in the WEKA data mining suite [19]. Naïve Bayes and C4.5 (decision tree) algorithms are used with default parameters while parameter k for the k-NN algorithm is set to 1 and parameter number of trees for the Random forest algorithm is set to 100. For SVMs we normalized data by setting normalize parameter. The parameter C for Linear SVM is set to 15. The parameters C, *degree*, *gamma* and *coefficient* for Polynomial SVM are set to 20, 3, 1 and 1, respectively. The parameters C and *gamma* for Radial SVM are set to 20 and 1, respectively.

3 RESULTS & DISCUSSION

Table 1 shows classification results of all ML algorithms. The accuracy of all ML algorithms is higher than 90% which suggests that the choice of our features for text/non-text differentiation is appropriate. The ML algorithm that has the highest accuracy 98.2% is the Radial SVM. The results obtained by the proposed text detection method are promising. Other authors who worked on similar problems also obtained comparable classification results. In [20] authors used SVM and classified text from non-text blocks with the accuracy of 96.62%. In [12] authors obtained 97.5% classification accuracy by using genetic programming to classify the document blocks as text, image, drawing and table. In [3] authors used Multilayer Perceptron to classify text from non-text blocks and obtained 97.25% of accuracy. The only disadvantage of approaches in [12, 3, 20] is that they fail to detect text in documents with complex layout, due to limitations of their segmentation algorithms. The advantage of our segmentation algorithm is that it self-adjusts to the document image regardless the image resolution and font size, it does not need any input parameters and it also works on documents with complex layouts. The only disadvantage is that it fails to detect (segment) some text blocks of very light text color, and also some text blocks with very complex (with a lot of details) background and very decorated text strings. In the future we plan to test our text detection method on other datasets and compare it with other segmentation methods.

Table 1: Classification results

Classifier	Naive Bayes	k-NN	C4.5 - decision tree	Random forest	Lin. SVM	Poly. SVM	Rad. SVM
Accuracy	90.1%	93.6%	94.3%	97%	97%	97.3%	98.2%
Precision of text blocks	0.860	0.911	0.937	0.976	0.964	0.961	0.978
Precision of non-text blocks	0.953	0.964	0.949	0.964	0.976	0.986	0.986
Recall of text blocks	0.958	0.966	0.950	0.964	0.976	0.986	0.986
Recall of non-text blocks	0.844	0.906	0.936	0.976	0.964	0.960	0.978

4 CONCLUSION

We have presented a text detection method, which consists of document segmentation and feature extraction algorithms. The proposed segmentation algorithm is based on the bottom-up strategy of analysis and segmentation is done by using the Sobel edge detection method. We created ASTRoID dataset of images, which consists of 500 image blocks of text and 500 image blocks of non-text. The proposed feature extraction algorithm extracts features from each image in ASTRoID dataset. We used 27 different features in order to differentiate text from non-text regions. In order to classify text from non-text blocks we used seven ML algorithms. The accuracy of all ML algorithms is higher than 90% which suggests that the choice of our features is appropriate. The classification results show that SVM with radial kernel has the highest accuracy 98.2%.

5 Acknowledgements

The presented work was supported by Creative Core FISNM-3330-13-500033 'Simulations' project funded by the European Union, The European Regional Development Fund. The operation is carried out within the framework of the Operational Programme for Strengthening Regional Development Potentials for the period 2007-2013, Development Priority 1: Competitiveness and research excellence, Priority Guideline 1.1: Improving the competitive skills and research excellence.

References

[1] Basilis, G.G. (2014). Imaging Techniques in Document Analysis Processes. In Doearmann, D. and Tombre, K. (Eds.) Handbook of Document Image Processing and Recognition (pp. 73–131). London: Springer.

[2] Barlas, P., Kasar, T., Adams, S., Chatelain, C. and Paquet, T. (2014). A typed and handwritten text block segmentation system for heterogeneous and complex documents. In: 11th IAPR International Workshop on Document Analysis Systems, pp. 46–50, IEEE Press, Tours, France.

[3] Bukhari, S.S., Azawi, M.A., Shafait, F. and Breuel, T.M. (2010). Document image segmentation using discriminative learning over connected components. In: 9th IAPR International Workshop on Document Analysis Systems, pp.183–190. Boston, MA, USA.

[4] Bukhari, S.S., Shafait, F. and Breuel, T.M. (2011). Improved Document Image Segmentation Algorithm using Multiresolution Morphology. In: 18th Document Recognition and Retrieval Conference, pp. 1–10., San Jose, CA, USA.

[5] Bukhari, S.S., Asi, A., Breuel, T.M. and El-Sana, J. (2012). Layout analysis for arabic historical document images using machine learning. In: Int. Conference on Frontiers in Handwriting Recognition, pp. 639–644.

[6] Burger, W. and Burge, M.J. (2009). Principles of Digital Image Processing. Springer, London.

[7] Coppi , D., Grana, C. and Cucchiara , R. (2014). Illustrations Segmentation in Digitized Documents Using Local Correlation Features. In: 10th Italian Research Conference on Digital Libraries, pp. 76–83. Procedia Computer Science, Vol. 38, Padua, Italy.

[8] Kise, K. (2014). Page Segmentation Techniques in Document Analysis. In Doearmann, D. and Tombre, K. (Eds.) Handbook of Document Image Processing and Recognition (pp. 135–175). London: Springer.

[9] Kruatrachue, B., Moongfangklang, N. and Siriboon, K. (2005). Fast Document Segmentation Using Contour and X-Y Cut Technique. In: The Third World Enformatika Conference. pp. 27–29. Turkey.

[10] Kundu, M.K., Dhar, S. and Banerjee, M. (2012). A New Approach for Segmentation of Image and Text in Natural and Commercial Color Document. In: Proc. of International Conference on Communication, Devices and Intelligent Systems, pp. 85–88. IEEE Press, India.

[11] Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics, Vol 9: 62–66.

[12] Priyadharshini, N. and Vijaya, M.S. (2013). Genetic Programming for Document Segmentation and Region Classification Using Discipulus. Int. Journal of Advanced Research in A.I., Vol. 2: 15–22.

[13] Priyanka, N., Pal, S. and Mandal, R. (2010). Line and Word Segmentation Approach for Printed Documents. International Journal of Computers and Applications, Vol. 1: 30–36.

[14] Roy, P.P., Pal, U. and Lladós, J. (2010). Touching Text Character Localization in Graphical Documents Using SIFT. In: Proceedings of the 8th International Conference on Graphics Recognition: Achievements, Challenges, and Evolution, pp. 199–211. Springer-Verlag, France.

[15] Shafait, F., Keysers, D. and Breuel, T. (2008). Performance evaluation and benchmarking of six-page segmentation algorithms. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 941–954. IEEE Press.

[16] Sumathi , C.P. and Priya , N. (2013). A Combined Edge-Based Text Region Extraction from Document Images. International Journal of Advanced Research in Computer Science and Software Engineering, 827–835.

[17] Vasuki, S. and Ganesan , L. (2005). Performance Measure for Edge Based Color Image Segmentation in Color Spaces. In: Proceedings of the International Conference on Emerging Technologies in Intelligent System and Control: Exploring, Exposing, and Experiencing the Emerging Technologies, pp. 621–626. Allied Publishers, Coimbatore.

[18] Vikas , J.D. and Vijay , H.M. (2011). Devnagari Document Segmentation Using Histogram Approach. Int. Journal of Computer Science, Engineering and Information Technology, Vol. 1: 46–53.

[19] WEKA (Open source, Data Mining software in Java) (2015). University of Waikato, New Zealand, http://www.cs.waikato.ac.nz/ml/weka [Accessed 02/06/2015]

[20] Zagoris, K., Chatzichristofis, S.A. and Papamarkos, N. (2011). Text Localization using Standard Deviation Analysis of Structure Elements and Support Vector Machines. EURASIP Journal on Advances in Signal Processing, Vol. 47: 1–2.