15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

Self-learning Structure for Text Localization

Supakorn Intaratat[1,2], Karn Patanukhom[1]
[1] Department of Computer Engineering, Chiang Mai University, Thailand
[2] Graduate School, Chiang Mai University, Thailand
supakorn_int@eng.cmu.ac.th, karn@eng.cmu.ac.th

Abstract

This paper presents a self-learning structure for text localization. The proposed system has an ability to improve itself automatically by analyzing unlabelled images. The system consists of three classification modules called component grader, component linker, and group classifier. Firstly, the image is analyzed to obtain the character candidate components. Then, the grader evaluates the possibility of text for every component by considering their properties individually while the linker classifies the degree of connection for every two components and groups all linked components together. Then, the groups of components are classified as text or non-text by the group classifier. Since all three modules work almost independently, we can update one module by using results from the other modules. This paper also presents a strategy for updating all modules by using unlabelled images. The experiment is given to show that the grader and the linker can be initialized by using few labeled training samples and then the system can automatically collect more samples from unlabelled images by using the results from three modules.

1. Introduction

Text detection or localization is a process to find text areas in the images that can be used in many applications such as in robot or image retrieval. The methods of text localization [1]-[8] can be categorized into region-based and texture-based approaches [6]. The regions-based methods [1]-[5] use a bottom-up process to detect text regions by grouping the pixels into regions based on their properties such as color or stoke width [4]. The advantage of the regions-based methods is robustness to the affine transformation. Maximally Stable Extremal Regions (MSER) [1] and Stroke Width Transform (SWT) [4] are examples of the region-based techniques. On the other hand, the texture-based methods such as Convolutional Neural Networks (CNN) [6], Gabor filters, or visual saliency map detect text regions by learning the differences between text regions and backgrounds. These methods are robust to blurs, text color inconsistency, shadows.

X. C. Yin, et al. [1] proposed a text detection method based on MSER and a forward-backward algorithm for detect multi-direction texts. The MSER algorithm is used for finding character candidates. The similar character candidates are grouped together into text candidates. The posterior probabilities and an AdaBoost classifier are used to elimination non-text.

Wiwatcharakoses, et al. [7] proposed a MSER based text localization method for multiple languages that robust to text orientation by using cascade decision chain and double threshold scheme to classify text regions from MSER components. The double threshold scheme is used to classify the text candidates into three classes that are high-confidence texts, low-confidence texts and non-texts then the final text detection results can be all high-confidence texts and some low-confidence texts that related or similar to high-confidence texts.

Huang et al. [8] developed the text detection scheme that takes advantage of both MSER and CNN. The MSER algorithm reduces the number of scanning window and increase ability of text low-condition detection. While the CNN algorithm with non-maximal suppression (NMS) can separates the multiple character connection and recover some missing characters.

However, to create text localization system, we need to prepare manually-labeled data to train the system. In this paper, we develop a self-learning structure for text localization problem that can reduce time and man-power consuming in the preparation process of training dataset. In addition, the proposed structure can also be used as transfer learning scheme for transferring some classification modules of the system that have learned for one language to be used for other languages.

2. The Proposed Self-learning Structure

An overview of the proposed scheme is illustrated in Fig. 1. The proposed scheme starts by extracting the possible regions that can be character candidates. This process can be implemented by MSER, SWT or other segmentation methods. Fig. 2 (b) shows an example of MSER-based candidate components obtained from the original image in Fig. 2 (a). The components obtained in this stage will include both texts and non-texts; therefore, we need to classify those components. Then, the system

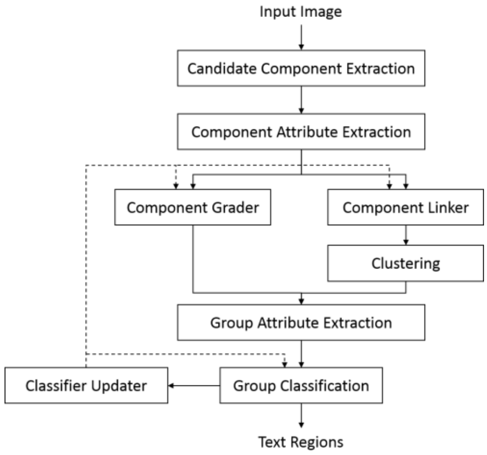


Figure 1. Structure of the proposed scheme.

extracts the necessary attributes or features for classification in the later steps. After that, every component will be graded by a "Component Grader" module. The components with higher possibility to be text are expected to have higher grader score. Fig. 2 (c) shows an example that visualizes the grader score of each component. Most of text components can obtain high grader score (red color) while most of non-text components obtain low grader score (blue color). The grader can be implemented by using any classification techniques such as probabilistic classifier, artificial neural network (ANN), or support vector machine (SVM).

The association between every pair of components is also analyzed by a "Component Linker" module. The relations of components are classified into three classes of connectivity that are L, G, and N as shown in Fig. 3. Class L is used to label any two adjacent components from the same text box. Class G is used to label any two non-adjacent components from the same text box. Class N is used to label any two unrelated components. The linker can also be implemented by using any classification techniques. After the image has analyzed by linker, all components are clustered into groups by considering classes of connectivity and orientations of text lines. Fig. 2 (f) shows an example of the clustering results where the components with same color are in the same group.

The final step is to classify the groups of components as text or non-text classes. In this scheme, attributes or features of each group are extracted by using the results of grader and linker. Any other features and classification techniques can be applied here. We can apply both the region-based features and the textural-based techniques such as GLCM [7], or CNN [6] in this step.

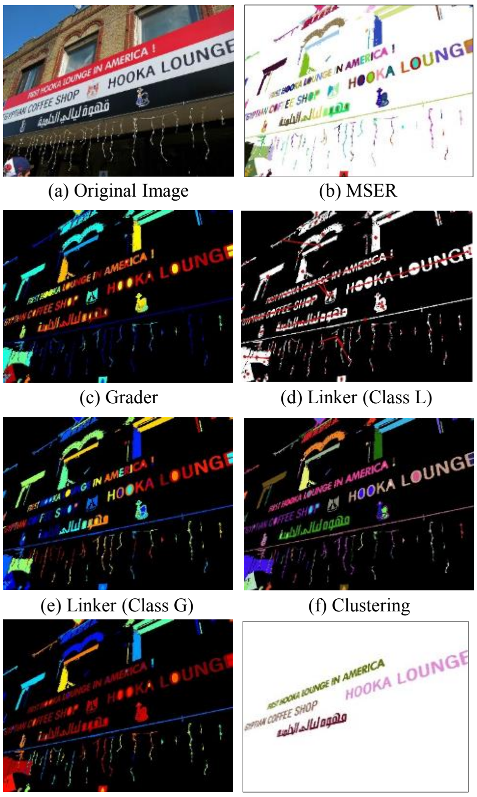
In conclusion, the proposed scheme consists of three classifiers (grader, linker, and group classifier) that work separately and use almost different features. The grader considers only the properties of individual component such as shape, size, or location while the linker considers only the mutual properties of the components such as differences in size, color, and stroke width. The group classifier collects the results from both grader and linker then extracted additional features to classify the clusters as text or non-text. The advantages of these three separated modules can be summarized as follows.

Transfer Learning: For example, if we have the system that trained for detecting English languages then we can create the new system for Burmese language by transferring the linker and the group classifier and using new grader that is trained for recognizing Burmese alphabets.

Self-learning: We can initialize the grader and the linker by using few labeled training samples. The system can automatically collect more samples from unlabelled images by analyzing the results from grader, linker and group classifier in a "Classifier Updater" module.

2.1. Candidate Components

In this paper, we implemented the MSER method [1] on the candidate component extraction process. The MSER can be extracted from a gray scale image by increasing intensity threshold step by step and comparing growth rate of each connected component in each threshold levels. We can remove the MSERs that are very small or very big or have high growth rate from the set of character candidates. We can also vary the number of MSERs by changing the step size of intensity threshold. The smaller step size will increase the number of MSERs that can improve the recall but reduce the precision of text extraction. The higher step size will decrease the number of MSERs that can improve the precision but reduce the recall. As a result, we apply the multiple-threshold scheme which the MSER components are firstly extracted by using very high intensity threshold to obtain the distinctive MSERs, and then the process is repeated by applying the lower intensity threshold. In this way, we can measure the quality of the MSERs by using their corresponding threshold levels.



(a) Original Image (b) MSER

(c) Grader (d) Linker (Class L)

(e) Linker (Class G) (f) Clustering

(g) Group Classification (h) Detected Texts

Figure 2. Example images in text detection process, (a) Original image, (b) Candidate components extracted by MSER, (c) Grader scores (blue = low value → red = high value), (d) Links of class L (red lines), (e) The number of links of class G connected to that component (blue = a few links connected→ red = many links connected) (f) Clustering results (same color = same cluster), (g) Group classification responses (blue = low possibility to be text→ red = high possibility to be text), (f) Final text mask.

After candidate components have been found, the necessary attributes for every component are prepared. These attributes will be used later in the classifiers. The attributes that we extracted in this paper are centroid position, colors, lengths of major and minor axes, area, aspect ratio, stoke width, MSER intensity threshold, the number of holes, solidity, extent, ratio of component perimeter to component area, the number of pixels located in the image border, centroid distance signature.

2.2. Component Grader

As mentioned in the previous section, function of the grader is to evaluate the possibility of each component to be a part of texts by considering properties of the component individually. The grader can learn by using a training set that consists of text and non-text samples.

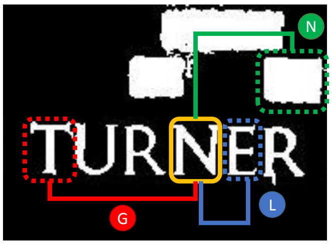


Figure 3. Classes of connectivity in component linker.

The training samples can be selected for some specific languages or for non-specific languages depending on the target application. In this paper, we implemented the grader by using the ANN where the responses from the output nodes will be used as the grader scores. If we use SVM or probabilistic classifiers as the grader then we can use the distances from the separated hyper-plane or posterior probability as the grader scores, respectively.

For the features, we can also select the set of features to classify some specific languages or non-specific languages depending on application. For example, if we want to detect the English texts then we can train the grader by choosing only the English alphabets for the positive training samples and using some effective OCR features such as CNN or Histogram of Oriented Gradient. Thus, we can expect that the grader will give high score for the components that look like the English alphabets. In addition, even though the initial training samples do not cover all alphabets or variation of font styles, the proposed system can obtain more training samples while analyzing new images and learn via classifier updater. On the other hand, if we want to create the system that is robust to language variation, the features should be less specific to the alphabet shapes to avoid the system over-fitting to any languages. Since, in this paper, we tested the proposed structure on multiple language dataset [7] therefore we selected only simple features for classification of non-specific language text and non-text components in the grader. The features used for grader are the number of holes and border pixels, aspect ratio, MSER intensity threshold, solidity, and perimeter ratio.

2.3. Component Linker

The linker is a module for classifying the connectivity of two components. To assign the classes of connectivity, the linker considers the differences in the attributes of two components. The results from linker can roughly identify text component by consider their connectivity. The components that are similar or link to many components tend to be text while the isolated components tend to be non-text as shown in Fig 2 (d)-(e). Most of attributes used in this paper are the attributes that are commonly used in other region-based methods [1], [7] such as spatial distance, stroke width variances, nearest neighbor ranking, differences in size, color, major/minor axis, and stroke width. Since the features for linker are almost completely different from the features for grader, these two modules can identify the text in different aspects. The linker can learn by any supervised methods using sample images that have extracted the components and classes of connectivity for every pair of components are labelled. In this paper, we also implemented the linker by using the ANN as in the grader.

After the classes of connectivity have been analyzed, the components are grouped together into candidate group. In this paper, we implemented the clustering process as in [7]. The clustering process is decomposed into two steps that are a single linkage clustering step and the line segmentation step. In the single linkage clustering step, every two components with link of class L will be assigned the same label of cluster. However, the results from single linkage clustering may contain the cluster with multiple text lines. The line segmentation step is done by changing class of the outlier links from L to G by consider distribution of spatial orientation of every links in the cluster and then re-labeling the clusters.

2.4. Group Attributes and Classification

In the final stage, the group candidates are classified to eliminate non-text groups. In this paper, we implemented group classifier by using the ANN with double threshold scheme as in [7]. Typically, the groups with most of components have high grader scores have high possibility to be text regions and the groups with most of components that have many links (class L and G) to the other components also have high possibility to be text regions. The average grader score and the average numbers of links of class L and G from the components in the group are used as group attributes for classification in this proposed structure. In addition, we also use the other features as in [7] that are the number of components, residual errors of polynomial regression, stroke width variance, average solidity, average extent, the average number of holes, maximum probability in histogram of link orientation, centroid distance signature, Haralick's features.

2.5. Classifier Updater

As mentioned in the previous section, after analyzing new images, the system can use the results to update all classification modules. The strategies for updating three modules are described as follows.

The grader can be updated by collecting samples from unlabelled images. We consider only the components in the groups that have moderate current group classifier responses. The positive samples can be collected from the components that have class-L connection to many components. The negative samples can be collected from the components that have very few class-L connections.

The linker can be also updated by collecting more samples from unlabelled images. We still consider only the components in the groups that have moderate current group classifier responses. The samples for the class L are obtained from the pair of components with high grader score that are currently classified as class L. The samples for the class G are obtained from the pair of non-adjacent components in the groups. The samples for the class N are collected from the pair of components in different groups that have very high current linker response of class N. Based on the preliminary experiment, it is more efficient to collect the addition samples for only class L and N.

Since the group classifiers used the responses from the grader and the linker as attributes for classification as mentioned in Section 2.4, when the grader and the linker are updated the distributions of grader/linker based group attributes may be changed. As a result, the group classifier should also be updated without changing training set. (only values of the features are updated).

3. Experimental Results

In this section, we conduct an experiment to demonstrate self-learning ability of the proposed scheme and compare accuracy of the proposed scheme to the other existing methods [3], [6], [7]. The results are evaluated in terms of precision, recall and F-score of the detected text boxes in comparison to the ground truth text boxes based on DetEval [10]. In this paper, the experiment is conducted on a multi-language dataset of 175 images which is the extended version of [7]. This test images consists of 1,239 text boxes from 10 languages such as English, Chinese, Japanese, Korean, Thai, and Arabic with various text directions and camera views.

At the first step, we initialize our proposed system by training the grader and the linker by using only one labeled image that consisted of two language, six text boxes, 33 text components, and 167 non-text components. Then, we train the group classifier by using the other 100 labeled images of various languages and text directions. After we have trained the initial grader, linker, and group classifier, we test the system on test set of 175 images. By varying the threshold of group classifier response, we can obtain the initial precision-recall curve as shown in Fig. 4 (Initial) where the best F-score is 27.7 as shown in Table 1.

Then, we let the grader and the linker automatically update by themselves. The additional training samples are automatically chosen from 175 unlabelled test images based on the strategy presented in Section 2.5. The group classifier is also updated by recalculating the values of the input features without changing training set. This version of updated system is named as "Updated-I". In this case, we let the proposed system performs self-learning from the test set and update itself before testing. The precision-recall curve is shown in Fig. 4 and its best F-score is 42.0 as shown in Table 1. The performance of the Updated-I is significantly improved from the initial system. The experimental results show that, by using the proposed scheme, the system can always learn and update itself when it is facing to new unknown images.

Finally, we try to let the grader and the linker perform self-updating by using 175 unlabelled test images plus 49 additional unlabelled images from BEST dataset [9]. This version of updated system is named as "Updated-II". The precision-recall curve is also shown in Fig. 4 and its best F-score is 42.6 as shown in Table 1. We can see that the performance of the system can still improve further when it learns from more unknown images.

We also compare the best F-scores of the proposed scheme with some existing methods as shown in Table 1 by using the same test set as the proposed approach. The proposed system (self-updated) can provide higher F-score than L. Neumann [3], T. Wang [6] and MSER (Baseline). By using the proposed scheme, we can obtain the system with self-learning ability by sacrificing only 1.6% of F-score from C. Wiwatcharakoses [7].

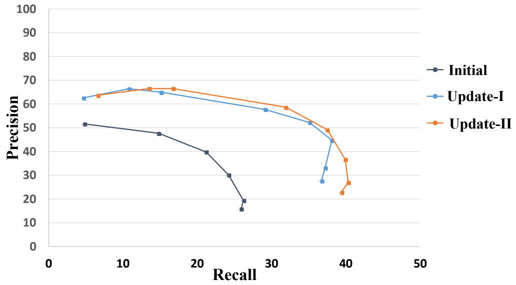


Figure 4. Precision-recall characteristics

Table 1. Performance comparison (The best F-score)

Methods	Recall	Precision	F-score
Proposed (Initial)	21.2	39.8	27.7
Proposed (Updated-I)	35.1	52.2	42.0
Proposed (Updated-II)	37.5	49.2	42.6
L. Neumann (Extremal Regions) [3]	30.9	34.5	32.6
T. Wang (CNN) [6]	32.6	39.1	35.6
C. Wiwatcharakoses (MSER) [7]	41.1	47.9	44.2
MSER (Baseline)*	12.7	19.3	15.3

*https://www.mathworks.com/help/vision/examples/automatically-detect-and-recognize-text-in-natural-images.html

4. Conclusions and Future Works

In this paper, we have developed the self-learning text localization system that can automatically improve itself by analyzing the unlabelled images as demonstrated in the experiment. For the future works, we plan to conduct more experiments on other datasets and study about other issues such as transfer learning ability and initialization effect. We also plan to improve the classifier updating strategy and apply incremental learning scheme.

References

[1] X. Yin, X. Yin, K. Huang and H. Hao, "Robust Test Detection in Natural Scene Images", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 36, No. 5, pp.970-983, May 2014.

[2] H. Chen, S.S. Tsai, G. Schroth, D.M. Chen, R. Grzeszczuk and B. Girod, "Robust text detection in natural images with edge-enhanced Maximally Stable Extremal Regions", 18th IEEE Inter. Conf. on Image Processing, pp.2609-2612, Sept. 2011.

[3] L. Neumann and J. Matas, "Real-Time Scene Text Localization and Recognition", 25th IEEE Conf. on Computer Vision and Pattern Recognition, pp. 3538-3545, 2012.

[4] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform", IEEE Conf. on Computer Vision and Pattern Recognition, pp.2963-2970, June 2010.

[5] C. Yao, X. Bai, W. Liu, Y. Ma and Z. Tu, "Detecting texts of arbitrary orientations in natural images", Computer Vision and Pattern Recognition (CVPR), 2012.

[6] T. Wang, David J. Wu, A. Coates, Andrew Y. Ng, "End-to-End Text Recognition with Convolutional Neural Networks", 2012 21st Inter. Conf. on Pattern Recognition (ICPR), pp.3304–3308, Nov. 2012.

[7] C. Wiwatcharakoses, K. Patanukhom, "MSER Based Text Localization for Multi-Language Using Double-Threshold Scheme", The 1st Inter. Conf. on Industrial Networks and Intelligent Systems (INISCom), pp.62-71, 2015.

[8] W. Huang, Y. Qiao, and X. Tang and X. Tang, "Robust Scene Text Detection with Convolution Neural Network Induced MSER Trees", European Conf. on Computer Vision (ECCV), 2014.

[9] National Electronics and Computer Technology Center, "Thai Character Recognition Contest", Benchmark for Enhancing the Standard of Thai language processing, 2014.

[10] C. Wolf and JM. Jolion, "Object count/Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms", Inter. Journal on Document Analysis and Recognition, pp. 280-296, 2006.