Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Science Letters:

Efficient page layout analysis on small devices*

Eun-jung HAN[†1], Chee-onn WONG[2], Kee-chul JUNG[†‡2], Kyung-ho LEE[1], Eun-yi KIM[3]

([1]*Department of Naval Architecture and Ocean Engineering, Inha University, Incheon 402-751, Korea*)

([2]*Human-Centered Interfaces Lab, Department of Digital Media, Soongsil University, Seoul 156-743, Korea*)

([3]*Internet and Media Engineering, Konkuk University, Seoul 143-701, Korea*)

[†]E-mail: hanej@inha.ac.kr; jungkeechul@gmail.com

Received Dec. 4, 2008; Revision accepted Mar. 25, 2009; Crosschecked Mar. 25, 2009

Abstract: Previously we have designed and implemented new image browsing facilities to support effective offline image contents on mobile devices with limited capabilities: low bandwidth, small display, and slow processing. In this letter, we fulfill the automatic production of cartoon contents fitting small-screen display, and introduce a clustering method useful for various types of cartoon images as a prerequisite stage for preserving semantic meaning. The usage of neural networks is to properly cut the various forms of pages. Texture information that is useful for grayscale image segmentation gives us a good clue for page layout analysis using the multilayer perceptron (MLP) based *x*-*y* recursive algorithm. We also automatically frame the segment MLP using agglomerative segmentation. Our experimental results show that the combined approaches yield good results of segmentation for several cartoons.

Key words: Efficient page layout analysis, MLP-based segmentation, Mobile devices, Image segmentation, Neural network
doi:10.1631/jzus.A0820842 **Document code:** A **CLC number:** TP391.4

INTRODUCTION

The popularity of mobile entertainment content improves with browsing techniques such as efficient page layout analysis (EPLA) on mobile devices. A review of EPLA shows that frame segmentation is the most important task in the entire process of browsing for proper identification. It is possible to browse the Web and media content. The performance evaluation of EPLA is to overcome these problems whereby offline image contents on mobile devices are usually produced by computer software such as Photoshop™. Normally, the image contents will be larger than what a mobile phone can hold. This is problematic for a mobile device with a medium to small sized screen and limited memory to host comic content as a form of mobile entertainment (Karlson *et al*., 2005). It is also difficult to create suitable contents with computers, since the cartoonists are accustomed to drawing cartoons by hand (Yin and Lee, 2004).

Our approach is to show how we obtained an EPLA that automatically produces the cartoon contents fitting for small screen, and to introduce a clustering method useful for various types of cartoon images as a prerequisite stage for preserving semantic meanings. In order to create automatically mobile cartoon contents fitting for mobile devices, splitting a cartoon and extracting texts are essential. Wang and Srihari (1989) used the above-mentioned approach to segment newspaper images into component regions, and Li *et al*.(2000) applied wavelet coefficient distributions to do top-down classification of complex document images. Etemad *et al*.(1997) adopted fuzzy decision rules for bottom-up clustering of pixels using a neural network. Many approaches to page segmentation concentrate on processing background pixels or using the 'white space' in a page to identify homogeneous regions. They can be regarded as top-down approaches that segment a page recursively by *x*- and

---

[‡] Corresponding author

[*] Project partially supported by the Ministry of Knowledge Economy (MKE) of Korea under the Information Technology Research Center (ITRC) Support Program, and the Basic Research Program of the Korea Science (No. R01-2006-000-11214-0)

y-cut from large components, starting with the whole page to small components, eventually reaching individual characters. It is not probable to directly adapt the methods that have been used for these relatively normal text documents to comic contents because the comic documents have much more diverse layouts (Chen *et al*., 2003).

We use an improved method to segment the offline contents frame using the multilayer perceptron (MLP) based *x*-*y* recursive algorithm to overcome the noise in the original offline contents, which is generated by the scanning process. The input of the neural network is a scanned image and the output is the candidate cutting points of the input image. In this method, several candidate cutting points are first generated by the *x*-*y* recursive cut algorithm, and we can identify whether the point indicates the right position using the MLP-based segmentation process on only candidate cutting points per step.

EPLA SYSTEM FOR MLP-BASED SEGMENTATION

The EPLA converts automatically the existing on/off-line cartoon contents into mobile contents using computer vision techniques. The EPLA at first will tentatively split a scanned image into frames, and then extracts the text regions before the image is minimized, since users cannot understand the excessively minimized texts. Based on the assumption that the text is located at the center of the balloon on white background, we extract the text using a connected components algorithm. First, to extract the text efficiently, we convert the split image into the binary image using thresholding. Then, we extract the text from the binary image using a connected component analysis. We locate the extracted text to the bottom of the screen. Through this process, we endow the consistency to each cartoon content, and can support the readability of cartoon contents. Lastly, we consider semantic regions of frames since they include important contexts of cartoon. Fig.1 shows the structure of EPLA.

For the page segmentation, at first, scanned images of the offline comic are used and changed into binary images for enhancing the processing speed. In a binary image of the input data, the black and white pixels are the recognized values in the MLP-based frame segmentation process. The input data make the cutting points with an MLP-based segmentation process. In this process, the MLP uses the weight calculated by the training process of some input images. The MLP finds the position of frame segmentation for the comic image. Then we can use the cutting point for the segmenting position and segment the frame using the *x*-*y* recursive concept. If the results have two or more candidate points, we could choose the right point by the verification process using the projection profile method. It is explained in the MLP structure and the cutting-point marking process (Fig.1).

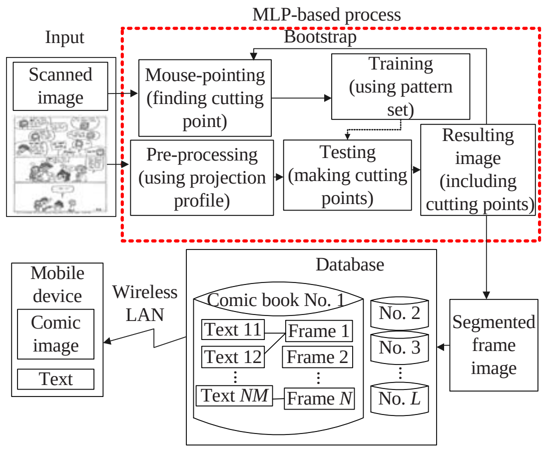


Fig.1 Overview of the proposed page segmentation approach

Pre-processing

We use the projection profile method for input images to produce input of the testing process. As shown in Fig.2, the histogram of the image is used to find the candidate cutting-point area using the loose threshold value. Then, the position from the *x*-*y* projection profile indicates the input to the testing process. This process has fewer input values than the whole input from the image that can be more efficient in terms of processing time.

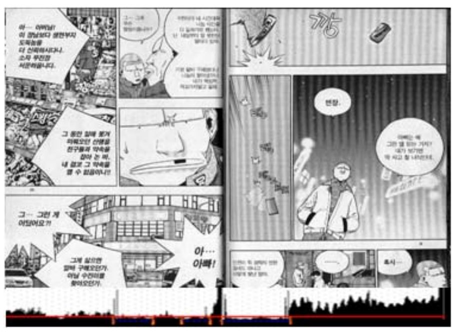


Fig.2 The input image area of testing process

Cutting-point marking

The structure of the MLP of our proposed design consists of 48 input nodes, 40 hidden nodes, and 1 output node. Fig.3 shows the structure of a two-layer neural network.

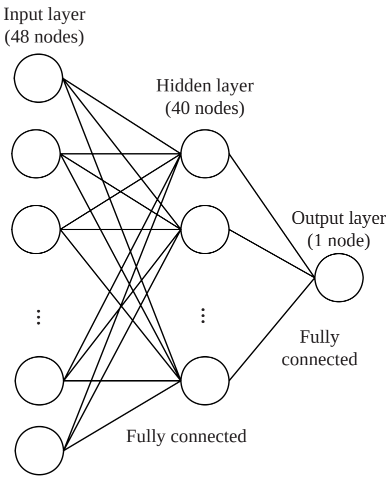
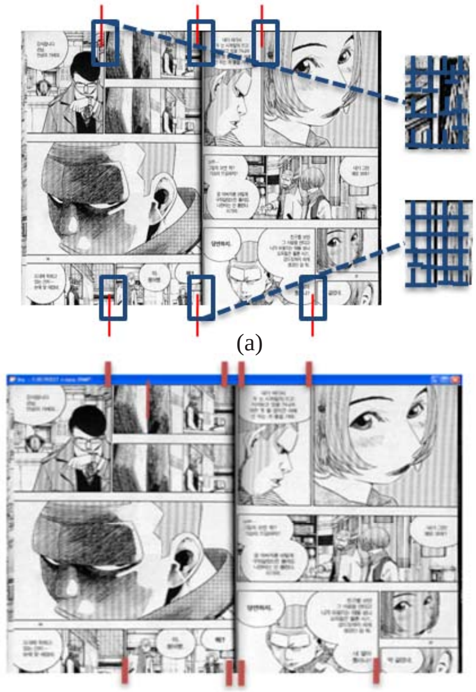


Fig.3 Structure of a two-layer neural network

The neural network has a fully-connected structure and uses a back-propagation learning algorithm. The MLP inputs a 48-order mesh vector to the network, which is extracted from a 30×40 normalized binary image. The 48 integer values are obtained by counting the number of pixels in each 5×5 local window in the normalized binary image. The resulting 25 intensities are normalized to the range of [0, 1.0]. Forty-eight floating point numbers are then input into the network in column major order (Bae *et al.*, 1998). If the position of the image is clicked by the mouse, the MLP recognizes its frame boundary to segment the image. A desired cutting point is determined manually and saved behind the 48-order mesh vector. Fig.4a shows the process of obtaining neural input values. The input values from the image are obtained to the boundary area of the images as a quadrangle from left to right and from top to bottom. The output value of the MLP is 1 or 0. In the forward process, the input image that is analyzed into 30×40 input pixels made by 48 nodes would have the MLP results. The true value indicates the frame boundary. If the area of the 30×40 input has a true value, the process can mark the cutting point. The position of the cutting point takes a role of segmenting points. If the result is false, the process can recognize that the area of the 30×40 input is not the frame boundary. The MLP can find the frame boundary and segment the frame. Fig.4b shows the cutting-point area.

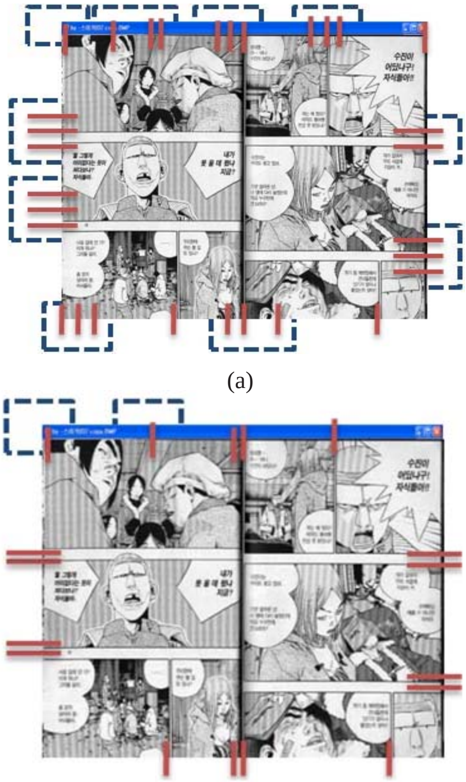


(a)

(b)

Fig.4 (a) The MLP input data of the comic image and zoomed in images; (b) The result of finding cutting points

The line of the frame boundary indicates the cutting points. The process can recognize the segmentation area with the MLP results. As can be seen in Fig.5, it is possible to segment artificially for offline images by training images. Although the frame has some noises, the MLP can recognize the frame boundary. Then we can mark the cutting points in the boundary and make the cutting line. However, the MLP result is not complete. As shown in Fig.6, if the object is near the frame boundary, the MLP process cannot recognize the frame boundary completely. To handle this problem we use the bootstrap method recommended by Sung and Poggio (1998), which was initially developed to train neural networks for face detection. Some non-frame samples are collected and used for training. In addition, the partially trained MLP is repeatedly applied to images for more complete segmentation, and then patterns with a positive output are added to the training set as non-frame samples. This process iterates until no more patterns are added to the training set (Jung, 2001).



(a)

(b)

Fig.5 Errors (marked by boxes on the boundary) on frame segmentation results
(a) The first forward image; (b) A result of the bootstrap method

Fig.6 One or more candidate cutting points

Verification

The output of MLP can have incorrect cutting points. In the forward process, the cutting line in the frame boundary shifts the position from top to bottom and from left to right. If the input image has two cutting points in the top and bottom, and it is projected to the same position, the forward process in MLP would also make the cumulative cutting points in the bottom and makes the cutting line. Then the candidate cutting points are more than required. Fig.6 shows this result. The dotted line of the image indicates the candidate cutting positions. Which one should be the segmenting point? To handle this problem, we use the projection profile method. For each cutting point, we can check the top-down pixels and find the real cutting point that is the segmenting position.

EXPERIMENT

This method was implemented in C++ programming language on an IBM PC. Two hundred images of offline comic were used to train MLP for the frame segmentation and the remaining 200 images were utilized for testing. The segmentation rates were evaluated using two metrics: precision and recall rate (Table 1). Eqs.(1) and (2) compute the precision (p) and recall rate (r), respectively:

$$p = \frac{\text{Number of correctly detected cutting points}}{\text{Number of detected cutting points}} \times 100\%, \tag{1}$$

$$r = \frac{\text{Number of correctly detected cutting points}}{\text{Number of desired cutting points}} \times 100\%. \tag{2}$$

Table 1 Comparison of precision and recall rate

Comic book type	Category	Precision (%)		Recall rate (%)	
		Without MLP	Proposed	Without MLP	Proposed
A	Training set	83.5	91.3	78	96.5
	Test set	−	87.7	−	92.6
B	Training set	81.5	90.3	76	95.5
	Test set	−	87.3	−	92.5
C	Training set	83.5	93.6	78	98.5
	Test set	−	87.2	−	92.5

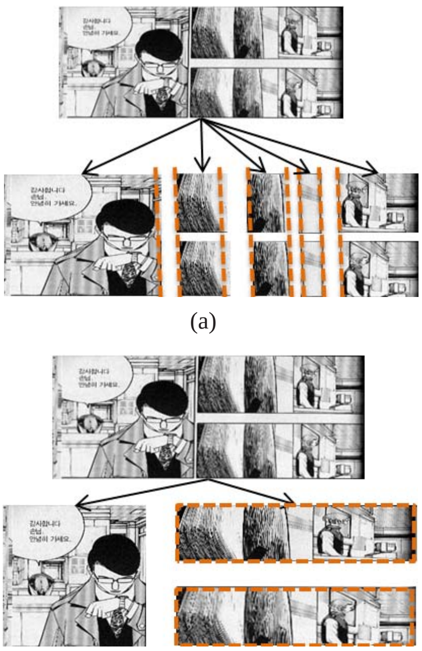
As shown in Table 1, our method produces higher precision and a better recall rate than the x-y recursive cut algorithm without an MLP process. However, it also shows a relatively low recall rate due to the lack of training data for comics A, B and C.

The execution with pre-processing is more efficient than that without pre-processing, because the sizes of input data from the comic are different (Table 2). By executing the pre-processing, we can reduce the input size. The method without MLP demonstrates very fast processing as compared to the proposed method; however, the precision and recall rates are low (Table 1). This proposed method using the MLP-based *x*-*y* recursive algorithm features a moderate speed and high precision.

Table 2 Comparison of execution time

Measurement	Execution time (s)
With pre-processing	2.8
Without pre-processing	10.0
Without MLP processing	1.5
Proposed method	2.8

The results of this algorithm are not exact what is shown in Fig.7a. Our new method using the MLP-based *x*-*y* recursive algorithm improves the frame segmentation accuracy with a variety of scanned images, which shows a transformation from large comic into segmented comic contents in a small screen mobile device. Fig.7b shows this result. Fig.7 shows the proposed frame segmentation result of mobile comic content that fits well on the mobile screen. The proposed method has an advantage in resizing the comic content depending on the mobile device screen.



(a)

(b)

Fig.7 (a) An *x*-*y* recursive algorithm result; (b) An MLP-based segmentation result

CONCLUSION

In this letter, we present an approach for page layout analysis using the MLP-based *x*-*y* recursive algorithm as a modality to effective segmentation on a small device. Our approach uses only the necessary transformations and techniques to adapt image contents to portable devices. The philosophy is to apply a set of transformations to the original image contents and multimedia contents in order to adapt them to the devices. This research examines closely the important prerequisite attributes of mobile contents service provider in terms of interaction and information utilization in the ubiquitous age.

Our future work includes automatic generation of cartoon contents fitting a small screen, and introduction of a clustering method useful for a variety of types of cartoon images as a prerequisite stage in preserving semantic meaning.

References

Bae, J.H., Jung, K.C., Kim, J.W., Kim, H.J., 1998. Segmentation of touching characters using an MLP. *Pattern Recogn. Lett.*, **19**(8):701-709. [doi:10.1016/S0167-8655(98)00048-8]

Chen, Y., Ma, W.Y., Zhang, H.J., 2003. Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices. Proc. 12th Int. WWW Conf., p.225-233.

Etemad, K., Doermann, D.S., Chellappa, R., 1997. Multiscale segmentation of unstructured document pages using soft decision integration. *IEEE Trans. Pattern Anal. Mach. Intell.*, **19**(1):92-96. [doi:10.1109/34.566817]

Jung, K.C., 2001. Neural network-based text location in color images. *Pattern Recogn. Lett.*, **22**(14):1503-1515. [doi:10.1016/S0167-8655(01)00096-4]

Karlson, A.K., Bederson, B.B., Giovanni, J.S., 2005. AppLens and LaunchTile: Two Designs for One-handed Thumb Use on Small Devices. Proc. SIGCHI Conf. on Human Factors in Computing Systems, p.201-210. [doi:10.1145/1054972.1055001]

Li, J., Gray, R.M., Olshen, R.A., 2000. Multiresolution image classification by hierarchical modeling with two-dimensional hidden Markov models. *IEEE Trans. Inf. Theory*, **46**(5):1826-1841. [doi:10.1109/18.857794]

Sung, K.K., Poggio, T., 1998. Example-based learning for view-based human face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **20**(1):39-51. [doi:10.1109/34.655648]

Wang, D., Srihari, S.N., 1989. Classification of newspaper image blocks using texture analysis. *Comput. Vision Graph. Image Processing*, **47**(3):327-352. [doi:10.1016/0734-189X(89)90116-3]

Yin, X., Lee, W.S., 2004. Using Link Analysis to Improve Layout on Mobile Devices. Proc. 13th Int. WWW Conf., p.338-344.