

# Text Segmentation from Images

Punam Patel  
M.tech(CSE)  
MITS,Laxmangarh

Shamik Tiwari  
Assistant professor  
MITS,Laxmangarh

## ABSTRACT

Texts detection from image or complex colored document is a very challenging problem. Text in images and videos contain useful information. There is a significant need to extract and analyze the text in images on Web documents, for effective indexing, semantic analysis and searching. The extraction of text information is very important because texts contain high-level semantic information. In this paper, we proposed a hybrid approach of text segmentation using edge and texture feature information. This result can also be used for other image interpretation and analysis.

## 1. INTRODUCTION

Texts in images and videos provide important information, which can help a machine to understand their content, retrieval and search. There are variety of approaches for text information extraction from images and video that have been proposed for specific applications including page segmentation, address block location, license plate location, text search in image, reading foreign language text and content-based image. It is not an easy task to design a general purpose text extraction system. This is because there are so many variations in images when extracting text from shaded or textured background, low contrast or complex images having variation in font color, size, style, orientation etc. These variations make the robust detection of text from these images a challenging problem. Two types of text may be present in an image or video frame, artificial or caption text which is artificially overlaid on the image, and scene text which exists naturally in the image.

Text segmentation from image is a very challenging problem. To design a general purpose text segmentation system for text, still an area of research. A lot of work has been done in text segmentation techniques.

Yu Zhong et al. [10] proposed a method to automatically localize caption text in JPEG compressed images. Caption text regions were segmented from background images using their distinguishing texture properties. They discussed a method which locates candidate caption text regions in the DCT compressed domain using the intensity variation information encoded in the DCT domain. Eun et al. [9] Proposed a method for automatic text location in images including multi-segment characters. They have used shape information in addition to size and location information. Michael et al. [7] Performed a detailed analysis of multilingual text characteristics, including English and Chinese. The text detection was carried out by edge detection, local thresholding, and hysteresis edge recovery. The coarse-to-fine localization scheme was then performed to identify

text regions. Xilin et al.[8] proposed the approach for text segmentation that embeds multi resolution and multi scale edge detection, color analysis for sign detection. They have used edge-based method for coarse detection accompanied by a multi-resolution scheme for different sign sizes.

Christos et al. [4] have shown a hybrid text localization approach for binarized printed document images. They have defined text localization processing using four processing stages: (i) Image binarization (ii) Pre-processing and Segmentation (iii) Feature extraction/feature selection (iv) Text localization. They have focused on pre-processing step and worked with both texture and spatial features. P.Rajendra et al. [2] proposed a method to enhance the quality of the image using DWT and IDWT, background removal in DCT domain and then texture feature computation on every 50\*50 block have been shown.

## 2. PROPOSED METHODOLOGY

Our method for text segmentation is a hybrid approach of segmentation using edge and texture feature information. The texture features such as homogeneity, contrast, energy for texts are different from non text. The texture features are used to detect the text region from image. The edge based textures have many desired properties. The gradient magnitudes usually have higher values in the edge of the characters, even when the text is embedded in pictures. Figure 1 shows the flowchart of the proposed methodology.

Algorithm based on edge and texture features:

Step 1: Conversion of color image to gray scale of image using,  $Y=0.299 * Red + 0.587 * Green + 0.119 * Blue$ .

Step 2: Edge detection is performed by 3\*3 sobel operator.

Step 3: A threshold is used for elimination weak edges.

Step 4: The edge image is divided into non overlapping blocks of m\*m pixels.

Step 5: Calculate the average magnitude per pixel and average gradient magnitude per pixel.

Step 6: Divide the filtered gray image into m\*m non-overlapping blocks. Here high pass filter is used to suppress background.

Step 7: Calculate the feature homogeneity and contrast at  $0^0$ ,  $45^0$ ,  $90^0$ ,  $135^0$  orientations for each block of original image using gray level co-occurrence matrix.

Step 8: Calculate the average of homogeneity and contrast for each blocks.

Step 9: Filter the text blocks using edge based feature and texture features.

Step 10: Merge the obtained text blocks.

## 2.1 Edge based feature

In the initial stage of algorithm we are interested in edges. One of the simplest ways for converting an image to gray-scale by  $Y=0.299 * Red + 0.587 * Green + 0.119 * Blue$ . The edge detection is then performed on the gray scale image by convolving the image with sobel masks, separately for horizontal and vertical edges. The sobel masking window are used because it is best suited to find the sharp edges of the text.

The Convolution is then performed by elimination of non-maxima and thresholding of weak edges. A threshold  $th_1$  is used for eliminating weak edges is computed using the formula:

$$th_1 = \sqrt{\frac{\sum_{i=1}^{h-1} \sum_{j=0}^{w-1} G_x^2(i,j) + G_y^2(i,j)}{(h-1)(w-1)}} \dots(1)$$

Where h and w are the height and width of the image and  $G_x(i, j)$  and  $G_y(i, j)$  are the x- and y-components of the gradient magnitude for a pixel at (i, j).

$G(i,j)$  can be calculated as,

$$G(i,j) = \sqrt{G_x(i,j)^2 + G_y(i,j)^2}$$

After this, the resulting image is divided into small non overlapping blocks of  $m*m$  pixels, where m depends on the image resolution. The value of m can be  $[0.05R \ 0.3R]$ , where R is the resolution of the image. For each block feature F is calculated using this formula:

$$F = \frac{\sum_{i=1}^{h-1} \sum_{j=1}^{w-1} G(i,j) * H(G(i,j) - th_1) * H(x(i,j) - 1)}{m^2} \dots(2)$$

Where  $H(i,j)$  is the edge image (1 – edge and 0-non-edge) and H is the step function.

$$H(x - a) = \begin{cases} 1 & \text{if } x \geq a \\ 0 & \text{if } x < a \end{cases}$$

In equation 2, F is the average gradient magnitude per edge pixel. The number of edges pixels and their gradient magnitude are usually higher for non texts blocks.

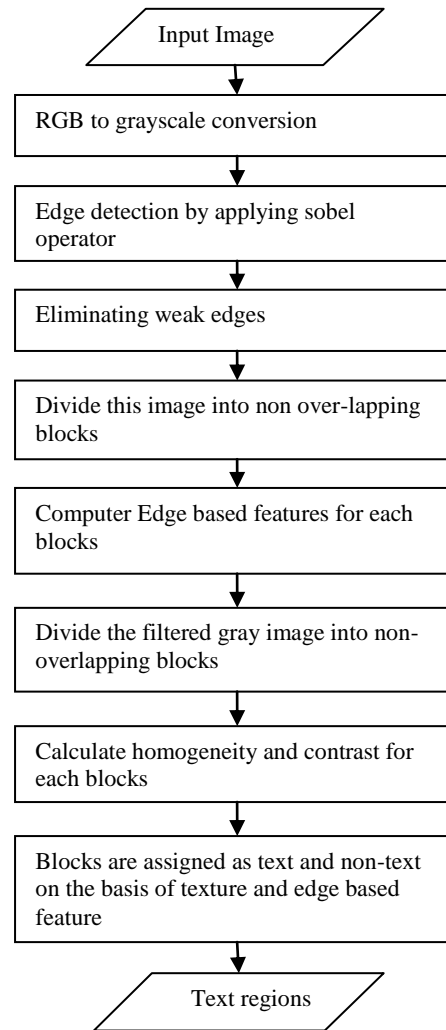


Figure 1: Flowchart of proposed method

## 2.2 Texture feature extraction

In this phase, the texture features such as homogeneity and contrast are obtained from every  $m * m$  blocks at  $0^0, 45^0, 90^0$ , and  $135^0$  orientations. Total 8 features are extracted from every block and are stored into two feature vector  $X_{i,j}$ (for homogeneity) and  $Y_{i,j}$ (for contrast), subscript  $i,j$  corresponds to  $(i, j)^{th}$  block. The feature vector for homogeneity and contrast is described as equations 3 and 4 :

$$X_{i,j} = [h1, h2, h3, h4] \dots(3)$$

$$Y_{i,j} = [c1, c2, c3, c4] \dots(4)$$

Where,

h1 and c1 corresponds to homogeneity and contrast at 0 degree orientation.

h2 and c2 corresponds to homogeneity and contrast at 45 degree orientation.

h3 and c3 corresponds to homogeneity and contrast at 90 degree orientation.

h4 and c4 corresponds to homogeneity and contrast at 135 degree orientation.

The features homogeneity and contrast are calculated as in equation 5 and 6.

$$\text{Homogeneity} = \sum_{i=1}^m \sum_{j=1}^m \left(\frac{P(i,j)}{R}\right)^2 \quad \dots(5)$$

$$\text{Contrast} = \sum_{n=1}^{m-1} n^2 \sum_{|i-j|=n} \left(\frac{P(i,j)}{R}\right) \quad \dots (6)$$

Where R is given in equation 7

$$R = \sum_{i=1}^m \sum_{j=1}^m P(i,j) \quad \dots(7)$$

P corresponds to co-occurrence matrix at a give degree.

R is normalized value of co-occurrence matrix P.

N is total number of blocks.

m \* m is dimension of block.

### 2.3 Block classification

We perform block classification using pre-defined threshold the  $th_2$  which will distinguish the text from the image (blocks  $F > th_2$  are assigned to the text and non text ). It is difficult to determine its value automatically  $F_{min}$  and  $F_{max}$  were found first. The value of  $th_2$  was required to be closer to lower bound so that no kind of text is missed. The value of  $th_2$  was determined using  $th_2 = k * (F_{max} - F_{min})$  where k varies between [0.1 0.3]. We assume that large value of F associated with text and small value of F associated with non text. The matrix  $L(i,j)$  decides whether the block belongs to text or not as equation 8.

$$L1(i,j) = \begin{cases} \text{text} & ; \text{ if } F(i,j) > th_2 \\ \text{non-text} & ; \text{ otherwise} \end{cases} \quad \dots(8)$$

Classification of text and non text blocks using texture feature require discriminant functions to classify every block into two classes  $w_1$  and  $w_2$  based on texture feature vector  $X_{(i,j)}$  and  $Y_{(i,j)}$ . Where,  $w_1$  corresponds to text blocks and  $w_2$  corresponds to non-text blocks category. The discriminant function uses two thresholds T1 and T2 corresponding to homogeneity and contrast values respectively. The discriminant functions  $d_1$  and  $d_2$  together decides a block as text block if the homogeneity and contrast features values at  $0^\circ, 45^\circ, 90^\circ$ , and  $135^\circ$  orientations are less then thresholds T1 and T2.

Given a feature matrix D of features  $X(i,j), Y(i,j)$  assign the corresponding image block to;

Class  $w_1$  : if  $d_1(X(i,j))$  is satisfied, and  $d_2(Y(i,j))$  is satisfied

Class  $w_2$  : otherwise

Where,

$d_1(X(i,j))$  is a dicriminant function which specifies constraint on homogeneity value.

$d_2(Y(i,j))$  is a dicriminant function which specifies constraint on contrast value.

$$d1(X(i,j)) = \begin{cases} \text{satisfied} & \text{if } X_{i,j}(h_j) \leq T1 \\ \text{not satisfied} & \text{otherwise} \end{cases} \quad \dots(9)$$

$$d2(Y(i,j)) = \begin{cases} \text{satisfied} & \text{if } Y_{i,j}(h_j) \geq T2 \\ \text{not satisfied} & \text{otherwise} \end{cases} \quad \dots(10)$$

Where ,

T1 corresponds to threshold on homogeneity

T2 corresponds to threshold on contrast.

### 2.4 Merging of text blocks to detect text regions

The merging process combines the text blocks connected in row and columns, to obtain new text regions  $r_{ij}$  if all the three equations 8,9 and, 10 satisfies. Binary image will be generated by marking the text region in white and non-text region in black. The result will be marked in original image.

$$r_{ij} = \begin{cases} \text{text} & ; (L1 = \text{text}) \ \&\& \ ((d1(X(i,j)) = \text{satisfied}) \ \&\& \ (d2(Y(i,j)) = \text{satisfied})) \\ \text{non-text} & ; \text{ otherwise.} \end{cases}$$

### 3. EXPERIMENTAL RESULTS

Figure 2 shows the various steps of segmentation method.



Figure 2.(a) Original image



Figure2.(b) Text regions are detected(white region)

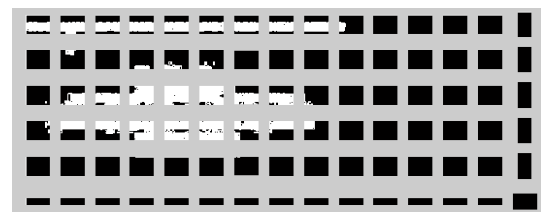


Figure 2.(c) Image is divided into non-overlapping blocks

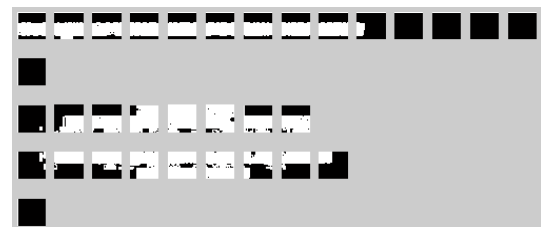


Figure 2.(d) Removal of non text regions

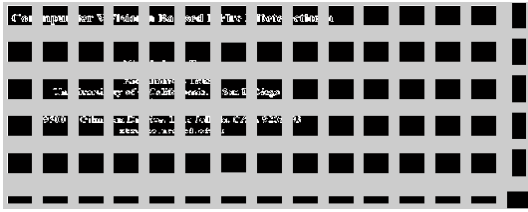


Figure 2.(e) Texts are shown in their respective blocks

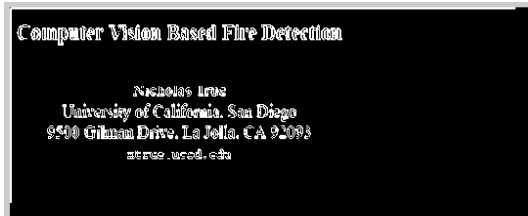


Figure 2.(f) Texts are detected

## 4. CONCLUSION

A set of images were selected with and without texts. The proposed method consists of two main phase of text extraction: strong edge extraction and texture feature extraction. Then, these images have been tasted using this hybrid approach of edge and texture features. The values of  $m$ ,  $k$  and threshold values of contrast and homogeneity were found through trial and error and also depends on resolution and quality of image.

## 5. REFERENCES

- [1] Adrian Canedo-Rodriguez, Jung Hyoun Kim, Soo-Hyung Kim , “Efficient Text Extraction Algorithm Using Color Clustering for Language Translation in Mobile Phone”, Journal of Signal and Information Processing, 3, 228-237, 2012.
- [2] P. Rajendra Kumar, Y. Rama Devi, T. Prathima, “Text Detection and Localization in Low Quality Video Images through Image Resolution Enhancement Technique”, International Journal of Computer Applications (0975 – 8887) Volume 58– No.6, November 2012.
- [3] Takahiro Shima , Kengo Terasawa and Toshio Kawashima, “Image Processing for Historical Newspaper Archives”, HIP, Beijing,China, 11september 16-september 17 2011.
- [4] Christos Emmanouilidis, Costas Batsalas and Nikos Papamarkos, “Development and Evaluation of Text Localization Techniques Based on Structural Texture Features and Neural Classifiers”, IEEE 10<sup>th</sup> international conference on document Analysis and Recognition 2009.
- [5] Cheolkon jung,Quifeng Liu , Joongkyu Kim, “ A new approach for text segmentation using a stroke filter ”, Science Direct, Signal Processing 88, 907–1916, 2008.
- [6] SungHoo Choi , Jong Pil Yun, KeunHwi Koo, JongHyun Choi, Sang Woo Kim, “Text Region Extraction Using Steel Making Process ”, 8th WSEAS International Conf. on Robotics, Control and Manufacturing Technology (ROCOM '08), Hangzhou, China, April 6-8, 2008.
- [7] Michael R. Lyu, Jiqiang Song and Min Cai, “A Comprehensive Method for Multilingual Video Text Detection, Localization, and Extraction”, IEEE Transactions on circuits and systems for video technology, Vol. 15, No.2, February 2005.
- [8] Xilin Chen, Jie Yang, Jing Zhang, and Alex Waibel , “Automatic Detection and Recognition of Signs From Natural Scenes”, IEEE Transaction On Image Processing,Vol. 13,No. 1,January 2004.
- [9] Eun Yi Kim , ,Jae Sik Chang and Hang Joon Kim, “ Automatic Text Location using Cluster-based Template Matching ”, IEEE ,1051-4651/02, 2002.
- [10] Yu Zhong and Hongjiang Zhang, “Automatic Caption Localization in Compressed Video”, IEEE Transaction on pattern analysis and machine intelligence,Vol.22, No.4,april 2000.
- [11] Rainer Lienhart and Wolfgang Effelsberg, “Automatic Text Segmentation and Text Recognition for Video Indexing”, ACM/Springer Multimedia Systems Magazine 1998.