www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume - 3 Issue - 8 August, 2014 Page No. 7583-7587

Extraction of text from images

Mamatha B S Chaithra B P
Department computer Science
PESCE
Mandya, India
Department computer Science
PESCE
Mandya, India

***Abstract*—Recent studies in the field of computer vision and pattern recognition show a great amount of interest in content retrieval from images and videos. This content can be in the form of objects, color, texture, shape as well as the relationships between them. The semantic information provided can be useful for content based image retrieval as well as for indexing and classification purposes. This text data is particularly interesting because text can be used easily and clearly describe the contents of an image. Since the data is embedded in an image or video in different font styles, sizes orientation, color and against a complex background the problem of extracting text becomes a challenging one. In this paper we present a method to extract text from images with complex background.**

Keywords-Text extraction, Edge detection, Text Localization

INTRODUCTION.

An image may be defined as a two dimentional function f(x,y) where x and y are spatial coordinates and the amplitude of f at any pair of coordinates (x,y) is called the intensity or grey level of the image at that point[1]. When x, y and the amplitude values of f are all finite, discrete quantities we call the image as digital image. The field of digital image processing refers to processing digital images by means of a digital computer. A digital image is composed of a finite number of elements each of which has a particular location and value. These elements are referred as picture elements, image elements and pixels.

Multimedia documents contain texts, graphics and pictures. Texts within an image play an important role in retrieval systems as they contain plenty of valuable information and can easily be extracted comparing with other semantic contents. Text with in an image is of particular interest as (i) it is very useful for describing the contents of an image (ii) it can be easily extracted with other semantic contents and (iii) it enables applications such as keyword based image search, automatic video logging and text based image indexing.

A variety of approaches to text information extraction from images have been proposed for specific applications including page segmentation, address block location, license plate location and content based image indexing. In spite of such extensive studies, it is still not easy to design a general purpose TIE system. This is because there are so many possible sources of variation when extracting text from a shaded or textured background, from low-contrast or complex images or from images having variations in font style, color, orientation and alignment.

In this paper we present a method to extract text from images with complex backgrounds. This is achieved using an edge-based text extraction algorithm based on the fact that edges are a reliable feature of text regardless of color/intensity, layout, orientations etc.

I. RELATED WORK

Various methods have been proposed in the past for detection and localization of text images. These approaches take into consideration different properties related to text in an image such as color, intensity, connected components, edges etc. These properties are used to distinguish text regions from their background and/or other regions within the image. Wang[4] proposed a connected component based method which combines color clustering, a black-adjacency graph an aligning and merging analysis scheme and a set of heuristic rules together to detect text in the application of sign recognition such as street indicators and billboards.

Lienhart[4] proposed a feed forward neural network to localize the segment text from complex images. Zhong[4] proposed a hybrid CC-based and texture based method to extract text. Kim combined a support vector machine(SVM) and continuously adaptive mean shift algorithm to detect and identify text region. Chen[4] also used a SVM to identify text lines from candidates. Gao[4] developed a three layer hierarchical adaptive text detection algorithm for natural scenes.

In this paper we use an edge-based text extraction algorithm which is robust with respect to font sizes, styles, color/intensity and complex image background.

II. SYSTEM OVERVIEW

Text extraction is the ability to extract text from an image. A Text Information Extraction (TIE) system receives an input in the form of a still image which can be in gray scale or color. This method includes three different stages:

Text detection, text localization and text extraction. Text detection refers to the determination of text in a given image. Text localization is the process of determining the location of text in the image. Text extraction is where text components are segmented from the background.

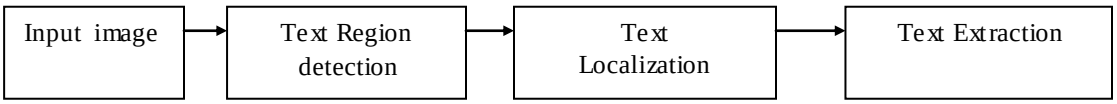


Fig1. Architecture of TIE system

III. EDGE BASED TEXT EXTRACTION ALGORITHM

The edge-based text extraction algorithm is robust with respect to font sizes, styles, color/intensity, effects of illuminations and complex image background. The algorithm based on the fact that edges are a reliable feature of text regardless of color/intensity, layout, orientations etc. edge strength, density and the variance of orientation are three distinguishing characteristics of text embedded in images which can be used as main features for detecting text. The algorithm consists of three stages, Text detection, text localization and text extraction.

A. Text Region detection

Given an input image, the region with a possibility of text in the image is detected. This stage aims to build a feature map by using three important properties of edges. Edge strength, density and variance of orientation. The resulting feature map is a grey scale image with same size of the input image where the pixel density represents the possibility of text.

Directional Filtering: In this stage, a Gaussian pyramid is created by successively filtering the input image with a Gaussian kernel of size 3x3 and down sampling the image in each direction by half. Down sampling refers to the process whereby an image is resized to a lower resolution from its original resolution. Each level in the pyramid corresponds to the input image at different resolution. Considering effectiveness and efficiency, four orientations 0^o, 45^o, 90^o, and 135^o are used to evaluate the variance of orientations where 0^o denotes horizontal direction, 90^o denotes vertical direction, 45^o and 135^o are the two diagonal directions respectively. A convolution operation with a compass operator results in four edge density images $E_{0=0,45,90,135}$ which contain all the properties of edges required in our method.

Edge selection: Vertical edges form the most important strokes of characters and their length also reflect the heights of corresponding characters. By extracting and grouping these strokes, we can locate text with different heights. However in a real scene under an indoor environment many other objects such as windows, doors, walls etc also produce strong vertical edges. Thus, not all vertical edges can be used to locate text. However, vertical edges produced by such non-character objects normally have very large lengths. Therefore, by grouping vertical edges into long and short edges, we can eliminate those vertical edges with extremely long lengths and retain short edges for further processing. After thresholding, these long vertical edges may become broken short edges which may cause false positives. In order to eliminate the false grouping caused by the broken edges, we used two stage edge generation method. The first stage is used to get strong vertical edges described in (1).

$$\mathbf{Edge}_{90}^{\mathbf{strong}} = |\mathbf{E_{90}}|_{\mathbf{z}} \quad (1)$$

where E_{90} is the 90^o intensity edge image which is the 2D convolution result of the original image with 90^o kernel, $|\bullet|_z$ is a thresholding operator to get a binary result of the vertical edges. Since this stage aims to extract the strong edges it is not very sensitive to the threshold value. In our implementation, we used Otsu's method to compute a global image threshold. The second stage is used to obtain weak vertical edges described in (2) to (4).

$$\mathbf{dilated} = \mathbf{Dilation}(\mathbf{Edge}_{\mathbf{90bw}}^{\mathbf{strong}})_{\mathbf{3x3}} \quad (2)$$

$$\mathbf{closed} = \mathbf{Closing}(\mathbf{dilated})_{\mathbf{mx1}} \quad (3)$$

$$\mathbf{Edge}_{\mathbf{90bw}}^{\mathbf{weak}} = |\mathbf{Edge}_{\mathbf{90}} \times (\mathbf{closed} - \mathbf{dilated})|_{\mathbf{z}} \quad (4)$$

where the morphological dilation with a rectangular structuring element of size 1×3 is used to eliminate the effects of slightly slanted edges and a vertical linear structuring element m×1 is then employed in a closing operator to force the strong vertical edges clogged. There is a trade-off on choosing the value m(i.e the size of structuring element). A small value costs less computation time at the expense of false positives. A large value increases the precise rate of detection but it increases computation cost as well. Considering the efficiency and effectiveness, from experiments, we found the value of m = (1 / 25) × width performs desirable detection results with an acceptable computation cost for a real time task. The resultant vertical edges are a combination of strong and weak edges as described in (5).

$$\mathbf{Edge}_{\mathbf{90bw}} = \mathbf{Edge}_{\mathbf{90bw}}^{\mathbf{strong}} + \mathbf{Edge}_{\mathbf{90bw}}^{\mathbf{weak}} \quad (5)$$

A morphological thinning operator followed by a connected component labeling and analysis algorithms are then applied on the resultant vertical edges as described in (6).

$$\text{Thinned} = \text{Thinning}(\text{Edge}_{90bw})$$

$$\text{Labeled} = \text{BWlabel}(\text{thinned}, 4) \quad (6)$$

where the morphological thinning operator makes widths of the resultant vertical edges one pixel thick. The connected component labeling operator (BWlabel) labels the thinned vertical edges. Here we used 4-neighbor connected component. After the connected component labeling, each edge is uniquely labeled as a single connected component with its unique component number. The labeled edge image is then processed by a length labeling process whose purpose is to let the intensity of edge pixels reflect their corresponding lengths. As a result, all the pixels belonging to the same edge are labeled with same nuber which is proportional to its length. Since a high value in the length labeled image represents a long edge, a simple thresholding described in (7) is used

$$\text{short}_{90bw} = |\text{Edge}_{90}^{\text{lengthlabeled}}|_z \quad (7)$$

where $|\bullet|_z$ is a "Binary-Inv" type of thresholding function. Since it is not easy to obtain 100% automatic correct detection . we attempt to minimize false negatives at the expense of false positives in our method. We use a low global threshold value and use edge density and variance orientation to refine them later on.

Feature map generation: As we mentioned before, regions with text in them will have significantly higher values of average edge density, strength and variance of orientations than those of non-text regions. We exploit these three characteristics to refine the candidate regions by generating a feature map which suppresses the false regions and enhances true candidate text regions. This procedure is described in (8) to (10)

$$\text{Candidate} = \text{Dilation}(\text{short}_{90bw}) \; \mathbf{m \times m} \quad (8)$$

$$\text{Refined} = \text{Candidate} \times \sum_{\theta=0,45,90,135} E_\theta \quad (9)$$

$$fmap(i,j) = N\left\{\sum_{m=-c}^{c}\sum_{n=-c}^{c}[refined(i+m, j+n)] \times weight(i,j)\right\} \quad (10)$$

where the morphological dilation with a m × m structuring element employed in the selected short vertical edge image(short_{90bw}) is used to get potential candidate text regions and the four orientation($E_{\theta=0,45,90,135}$) is used to refine the potential candidate regions. In above uations , fmap is the output feature map and N is a normalization operation that normalizes the intensity of feature map into a range of [0, 255]. Here weight(i, j) is a weight function which determines the weight of pixel (i, j) based on the number of orientations of edges within a window. Namely, the more orientations the window has the larger weight the central pixel has. By using weight function, our method distinguishes the text regions from texture-like regions, such as window frames, wall patterns etc.

B. Text Region localization

The process of localization involves further enhancing the text regions by eliminating non-text regions. This stage localizes text regions through two steps: feature clustering, heuristic filtering.

Feature clustering: Since the intensity of the feature map represents the possibility of text, a simple global thresholding can be employed to highlight those with high text possibility regions resulting in a binary image. Normally, text embedded in landmarks such as nameplates and information signs appears in clusters i.e it is arranged compactly. Thus, characteristics of clustering of clustering can be used to localize text regions for indoor mobile robot navigation. A morphological dilation operator can easily connect the very close regions together while leaving those whose positions are far away to each other isolated. In our method, we use a morphological dilation operator with a 7x7 square structuring element to the binary image obtained from the previous step to get joint areas referred to as text blobs.

Heuristic filtering: In order to be seen easily by human, characters in information signs and nameplates usually are either large enough or small but appears in clusters. As a result, the area as well as the width and height ratio of each text blob cannot be too small. In our method, two constraints are used to filter out those blobs which do not contain text. the first constraint is used to filter out all the very small isolated blobs described in (11) where it uses a relative area value instead of the absolute value for each blob. Therefore, there is no specific limitation for the font size, which makes more sense for the real world scenes.

$$\text{Area}_{\text{region}} >= \mathbf{(1/20)\ x\ Area_{max}} \quad (11)$$

The second constraint described in (12) filters out those blobs whose widths are much smaller than corresponding heights. A threshold of "0.2" is considered for the worst case, such as single western letter "I" or Arabic digit "1".

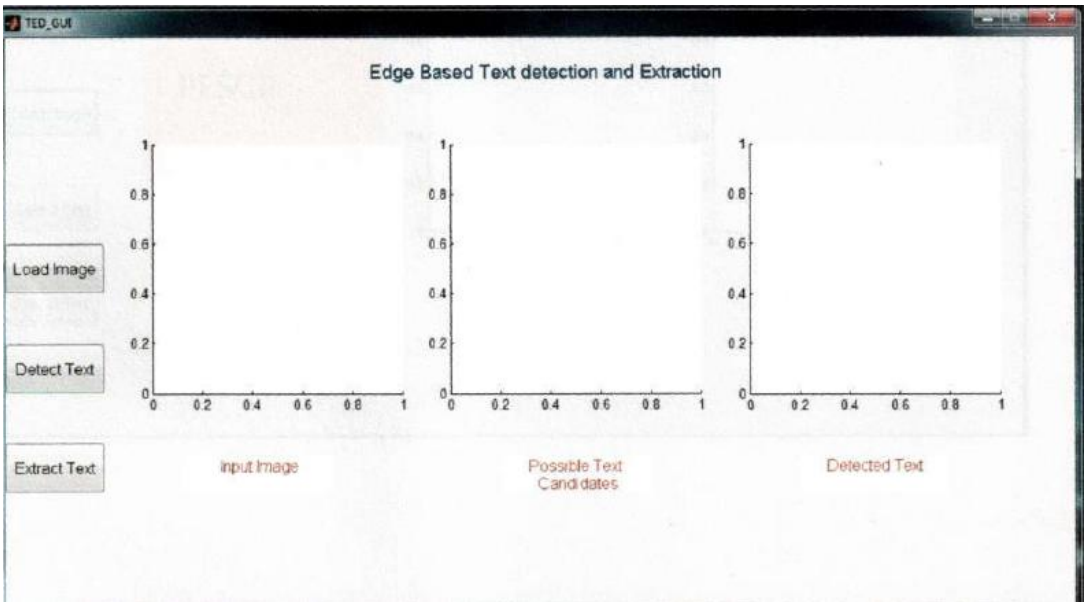
$$\text{Ratio}_{w/h} = \frac{\text{Width}_{\text{region}}}{\text{Height}_{\text{region}}} >= 0.2 \quad (12)$$

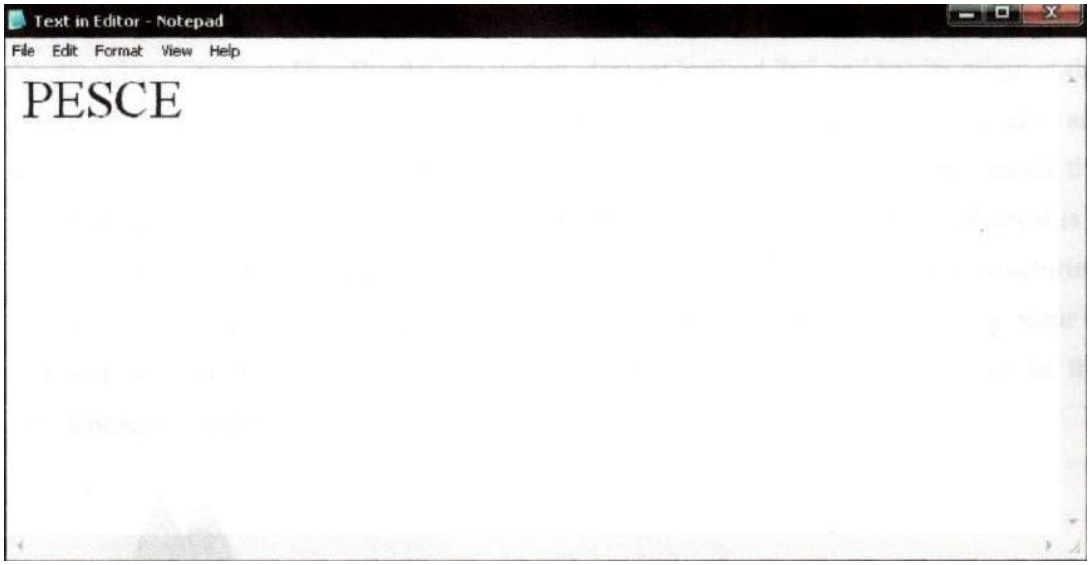
C. *Character extraction*

Existing OCR(Optical Character Recognition) engines can only accept simple inputs i.e it only deals with printes characters against clean backgrounds. The purpose of this stage is to extract accurate binary characters.

IV. EXPERIMENTAL RESULTS

We have applied our edge based text detection algorithm on a set of images with different font sizes, perspective and under different lighting conditions. The algorithm is implemented using MATLAB functions. A graphical user interface is also developed using MATLAB GUI components. The sample test image with expected and observed output for the image with text "PESCE" is shown below.





V. CONCLUSION

We have used an edge-based text extraction algorithm which can localize text from images. Experimental results show that this method is very effective and efficient in localizing and extracting text-based features. Our method is robust with respect to font sizes, styles, uneven illuminations and reflection effects. Future work can include combining the edge and connected component based algorithms, morphological cleaning of images and etc.

References

[1] Rafael.C.Gonzalez, Richard.E.Woods and Steven.L.Eddins, Digital image processing using MATLAB.

[2] Keechul Jung, Kwang In Kim, Anil k Jain, Text Information extraction in images and Video. A survey, The journal of the Pattern recognition soceity,2004

[3] Xiaoqing Liu and Jagath Samarbandu, Multiscale Edge-based text extraction from complex images.

[4] Jagath Samarbandu,, Member IEEE and Xiaoqing Liu, Text extraction for Mobile robot navigation.

[5] J.Sushma, Dept of E&C, Text detection in color images.