International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 11 | Nov 2018 www.irjet.net p-ISSN: 2395-0072

Devnagari Text Detection

Anugrah S[1], A. Sanghi[2], A. Shukla[3], R. Chaturvedi[4]

[1,2,3,4]*Computer Science, Army Institute of Technology, Pune University*

---***---

Abstract - *In this article, we present a robust scheme for detection of Devanagari texts in scene images. These are the two most popular scripts in India. The proposed scheme is primarily based on two major characteristics of such texts - (i) variations in stroke thickness for text components of a script are low compared to their non-text counter- parts and (ii) presence of a headline along with a few vertical downward strokes originating from this headline. We use the Euclidean distance transform to verify the general characteristics of texts in (i).*

Key Words: Text recognition, Devnagari, connected components extraction, computer vision.

1.INTRODUCTION

Detection of texts in images of natural scenes has enough application potentials. However, related studies are primarily restricted to English and a few other scripts of developed countries. Two surveys of existing methods for detection, localization and extraction of texts embedded in images of natural scenes can be found in [1] . A few of the recent studies on the problem include [3] In the Indian context, there are often texts in one or more Indian script(s) in an image of natural outdoor scenes. Devanagari and Bangla are its two most popular scripts used by around 500 and 220 million people respectively. Thus, studies on detection of Devanagari texts in scene images are important. In a recent study, Bhattacharya et al.[11] proposed a scheme based on Roy Chowdhury, Bhattacharya and Parui morphological operations for extraction of texts of these two scripts from scene images.

Existing approaches for text detection can be broadly categorized into connected component (CC) based and texture based algorithms. The CC based methods are relatively simple, but they often fail to be robust. On the other hand, although texture-based algorithms are more robust, they usually have higher computational complexities.

A well-known feature that text components have approximately uniform stroke widths throughout a character or letter unlike most other components present in a scene image, has been used before . In [8], an input image is scanned horizontally to identify pairs of sudden intensity changes and the intermediate region is verified for approximate uniformity in color and stroke widths. The limitations of the approach in [8] have been described in [9]. In this later work certain Stroke Width Transform (SWT) was designed based on the Canny image [12] by following rays along the gradient direction of an edge pixel to reach to another edge pixel roughly opposite to the former one. The distance between them was used to assign the stroke width of each pixel along the path of traversal.

As a solution to this problem, we use the well-known distance transform [13] for detection of candidate text regions and the detail of our strategy for the same is described in Section 3.2. In Section 3.3, we define a set of general rules based on the geometry of text regions for elimination of some of the false positive. responses of the scheme described in Section 3.2. At the end of this stage, texts of non-Indic scripts should also get selected. Presence of headline, a characteristic feature of Devanagari texts, is verified next and its computation based on probabilistic Hough line transform [14] is presented in Section 3.4. In the earlier work [11], morphological operations were employed for detection of headline of and Devanagari texts. However, this approach fails when such texts are sufficiently inclined. In the proposed strategy, the above problem is solved by using probabilistic Hough line transform for the purpose of detection of prominent lines in the image. Subsequent use of script specific characteristics helps to identify the presence of headline in candidate text regions.

Fig -2: Street boards in India.

2. DEVNAGARI TEXT CHARACTERISTICS

There are 50 basic characters in the alphabets of Devanagari scripts. For both these scripts, often two or more consonants or one vowel and one or two consonants combine to form different shapes called compound characters. Devanagari have a large number of such compound characters. Additionally, the shapes of the basic vowel characters (except the first one) get modified when they occur with a consonant or a compound character. The shape of a few basic consonant characters also gets modified in a similar situation. Most of the characters of both scripts have a horizontal line at their upper part. This line is called the headline. In a continuous text of these scripts, the characters in a word often get connected through this headline.

A text line of any of these two scripts has three distinct horizontal zones. The portion above the headline is the upper zone and below it but above an imaginary line called the baseline, is the middle zone while the part below the baseline is called the lower zone. There are many vertical segments in the middle zone of Devanagari texts.

3. PROPOSED WORKING

In a previous study [11], we observed that binarization of scene images often results in partial or complete loss of textual information. However, connected component (CC) analysis based on Canny edge detector has less number of cases of low-contrast regions being missed out. In the present work, we studied a robust scheme for finding CCs from Canny image along with a few rules for detection of Devanagari components.

Fig -3: Input image and devnagari text detection.

3.1 Preprocessing and Connected Components

An input color image (I) is first converted to 8-bit grayscale image (G). We use Canny operator [12] to get the edgemap (E) from G. This step is perhaps the most critical towards the success of the proposed approach and a brief description of our present implementation is provided. The Canny edge detector in OpenCv has three parameters - val1, val2 and val3. We used val3 = 3 for Gaussian smoothing of the input image with 3 3 kernel, the Gaussian being determined using window-size (wx= 3, wy = 3)

The larger of val1 and val2 is used as a threshold for selection of prominent edges and the smaller of these two is used as a distance threshold for linking of nearby edges. On the basis of the training samples of our database of scene images, we selected val1 = 196 and val2 = 53. This value of val2 helped us to avoid linking of edges of text components with edges of background objects. On the other hand, such a choice of val2 often leaves edges of a text component segmented into smaller pieces. We solved this problem by applying a morphological closing operation with a 3 3 kernel anchored at center on E as a post-processing operation of the Canny edge detector. This often helps to connect broken edges of the same character or symbol. Also, many erratic edges of background objects merge to form a larger component.

For further analysis, we consider the smallest bounding rectangle S in the image G corresponding to each connected component obtained by the above operations.

Fig -4: Preprocessing and CC extraction. Input image and inverted image.

Fig -5: Local thresholding and inversion.

Fig -6: Morphological closing, skeleton of image and morphological closing on skeleton for line detection.

3.2 Extraction of stroke width

Each sub-image S obtained in Section 3.1 is binarized and subjected to the Euclidean distance transform (DT) [13]. Each pixel in the resulting image is set. to a value equal to its distance from the nearest background pixel. Thus, we compute the distance of each object pixel from its edge or boundary.

A. Determination of Background Color

Texts can appear lighter against dark background or darker against light background. In [9], the distance between edges of opposing gradients was computed along both +ve and -ve gradient directions to account for both the possibilities of lighter or darker texts. In the proposed scheme, we consider the sub-image S and its inverse S and compute the DT for each of them as shown below. Let the corresponding transformed images be D and D . Now, we compute the number of zeros as well as the number of non-zeros along the four boundaries of both D and D . The number of zeros will be larger for a sub-image with lighter foreground against dark background and the corresponding DT (D or D) is selected as D.

Some letters may be so aligned that they have majority object pixels present along boundaries, giving a wrong estimation of background color. To deal with this, instead of using the minimum bounding rectangle of each component we increase its size by adding a small integer m (in our implementation, m = 2) to its dimensions, taking care of image boundary overflows.

Thus, a larger portion of background pixels is sampled in the bounding rectangle defining the sub-image with fewer chances of foreground pixels being wrongly counted while checking border pixels.

It is to be noted, for the purpose of background color estimation, that even a binarized image would have sufficed. However, as the distance transform is required for subsequent stroke thickness calculation also, we do not perform the extra step of thresholding.

B. Determination of Stroke Thickness

For each pixel with non-zero D, we consider a 3x3 window centered at the pixel. If the D value of the pixel is a local maximum among the nine such values, we store the D value in a list < T > for further processing. Such a D value (a local maximum value) is an estimate of half of the local stroke thickness. Finally, we compute the mean and the standard deviation of the local stroke thickness values stored in < T >. If > 2 (well-known 2 − limit used in statistical process control), we decide that the thickness of the underlying stroke is nearly uniform and select the sub-image S as a candidate text region.

Fig -7: Detection based on headline and vertical line.

3.3 Determination of Headline in Devnagari Text

In order to identify regions of Devanagari texts from among the regions in the set < V >, we compute a few common characteristics features of these two scripts as described below. In each of the above regions we compute the progressive probabilistic Hough line transform (PPHT)[14] to obtain the characteristic horizontal headlines of Devanagari texts. This trans- form usually results in a large number of lines and we consider only the first n prominent (with respect to the number of points lying on them) ones among them. A suitable value of n is selected empirically. Now, the lines with absolute angle of inclination with the horizontal axis less than (selected empirically to allow significantly tilted words) are considered as horizontal lines. A necessary condition for selection of a member of < V > as a text region is that these horizontal lines appear in its upper half. Let < L > denote the set of such horizontal lines corresponding to a region.

3.4 Using similarity methods for detecting missed out text region

The main criterion used in the above for selection of texts of Indian scripts is the presence of a headline, which in turn depends on the Hough transform being able to pick up the headline and the vertical strokes immediately below the headline. There are several cases where the headline may be too small and also there are certain situations where it does not occur at all. To detect possible Devanagari text regions in < V > < M >, which do not exhibit the headline property as in the above, we recursively loop through the regions of < M > and shift a member of < V > < M > to < M > provided it has high similarity with one of the current members of ¡ M ¿ with respect to its height, width, relative position and average stroke thickness. We stop when no addition is made to the current list of < M >. Values of parameters involved in these similarity measures are decided empirically.

4. RESULTS

We tested the algorithm on a sample data set of 10,000 diverse images which were of different qualities and of different camera angles. We found that our algorithm was able to recognise devanagari script with a precision of 0.7994, recall of 0.778, f-measure of 0.784. This is an improvement over (paper by Bhattacharya et al[11]). We also found that our algorithm was able to recognise devanagari script where the image itself was obscured through markings and printing mistake. We also were able to develop our algorithm in a way that the english script was completely ignored if present.

5. CONCLUSION

Although the simulation results of the proposed method on our image database of outdoor scenes containing texts of major Indian scripts are encouraging, in several cases, it produced false positive responses or some of the words or a part of a word failed to be detected. Another major concern of the present algorithm is the empirical choice of a number of its parameter values. We are at present studying the effect of using machine learning strategies to avoid empirical choice of the values of its various parameters. Preliminary results show that this will improve the values of both p and r by several percentages. However, we need more elaborate testing of the same. In future, we plan to use a combined training set comprising of training samples from both of our and the ICDAR 2003 image databases so that the resulting system can be used for detection of texts of major Indian scripts as well as English. Finally, identification of scripts of detected texts is necessary before sending them to the respective text recognition modules. There are a few works [17] in the literature on this script identification problem. Similar studies of script identification for texts in outdoor scene images will be taken care of in the near future.

ACKNOWLEDGEMENT

We thank Dr.Jayadevan.R for providing us with this opportunity to work on his dataset. We extend our acknowledgement to Prof. Asha Sathe, overall-in-charge, Prof. S. Dhore HOD Computers, Prof. M. B Lonare and all other staff and members of Computers AIT Pune

REFERENCES

[1] Liang, J., Doermann, D., Li, H.: Camera Based Analysis of Text and Documents : A Survey. Int. Journ. on Doc. Anal. and Recog. 7, 84104 (2005)

[2] LJung, K., Kim, K. I., Jain, A. K.: Text Information Extraction in Images and Video: a Survey. Pattern Recognition 37, 977997 (2004)

[3] Li, H., Doermann, D., Kia, O.: Automatic Text Detection and Tracking in Digital Video IEEE Trans. Image Processing 9, 147167 (2000)

[4] Gllavata, J., Ewerth, R., Freisleben, B.: Text Detection in Images Based on Unsupervised Classification of High Frequency Wavelet Coefficients. Proc. of 17th Int. Conf. on Patt. Recog. . 1, 425428 (2004)

[5] Saoi, T., Goto, H., Kobayashi, H.: Text Detection in Color Scene Images Based on Unsupervised Clustering of Multichannel Wavelet Features. Proc. of 8th Int. Conf. on Doc. Anal. and Recog . 690694 (2005)

[6] Ezaki, N., Bulacu, M., Schomaker, L.: Text Detection From Natural Scene Images: Towards a System for Visually Impaired Persons. Proc. of 17th Int. Conf. on Patt. Recog. II 683686 (2004)

[7] LJung, K., Kim, K. I., Jain, A. K.: Text Information Extraction in Images and Video: Image and Vis. Comp. 23, 565576 (2005)

[8] Subramanian, K., Natarajan, P., Decerbo, M., Castanon, D.: Character-Stroke Detection for Text-Localization and Extraction Proc. of Int. Conf. on Doc. Anal. and Recog. 3337 (2005)

[9] Epshtein, B., Ofek, E., Wexler, Y.: Detecting Text in Natural Scenes with Stroke Width Transform. Proc. of Int. Conf. on Doc. Anal. and Recog. 3337 (2005)

[10] Kumar, S., Perrault, A.: Text Detection on Nokia N900 Using Stroke Width Transform. http://www.cs.cornell.edu/courses/cs4670/2010fa/ projects/ final/results/group of arp86 sk2357/Writeup.pdf

[11] Bhattacharya, U., Parui S. K., Mondal, S.: Devanagari and Bangla Text Extraction from Natural Scene Images. 10th Int. Conf. on Doc. Anal. and Recog. 171175 (2009)

[12] Canny, J.: A Computational Approach to Edge Detection. Patt. Anal. and Mach. Intell. 8 679714 (1986)

[13] Borgefors, G.: Distance Transformations in Digital Images. Comp. Vis., Graph. and Image Proc. 34 344371 (1986)

[14] Matas, J., Galambos, C., Kittler, J.: Progressive Probabilistic Hough Transform. Progressive Probabilistic Hough Transform. Proc. of BMVC98. 1 256265 (1998)

[15] Bradski, G., Kaehler, A.: Learning OpenCV. 2008 OReilly Media, Inc., 2008

[16] Lucas, S. M. et al. ICDAR 2003 Robust Reading Competitions. Proc. of 7th Int. Conf. on Doc. Anal. and Recog. 682668 (2003).

[17] Zhou, L., Lu, Y., Tan, C.L.: Bangla/English Script Identification Based on Analysis of Connected Component Profiles. Proc. Doc. Anal. Syst. 243-254 (2006)