ISSN:0975-9646

Jenil Shah et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (5) , 2014, 6827-6830

A Simple and Effective Optical Character Recognition System for Digits Recognition using the Pixel-Contour Features and Mathematical Parameters

Jenil Shah[#1], Viral Gokani[*2]

[#]*Department of Electronics & Communication, Dharmsinh Desai University College Road , Nadiad – 387001 , Gujarat , India*

[*]*School of Electronics Engineering, VIT University, Chennai Campus Vandalur - Kelambakkam Road, Chennai – 600127, Tamilnadu, India*

***Abstract*— This paper presents a simple and effective Optical Character recognition system (OCR) for accurate detection of digits. Initially 0-9 digits are classified into four groups using the background pixel range. After detection of the group, the digits are distinctly identified using intrinsic ratio as mathematical parameter. Furthermore any overlapping digits are recognized using curved contour coordinates. Experimenting with a large data set we have extracted the exact range of all the parameters used for recognition. Recognition results and lucid flow reveals simplicity of the algorithm.**

Keywords*— Simple OCR , Digit recognition , Digit OCR , OCR Algorithm

I. INTRODUCTION

With the advancement in technology and processing speed, more and more complex algorithms for Optical Character Recognition system involving machine learning and neural networks are proposed. OCR is a process of converting an image representation into editable and text format. It is the method for digitizing the printed and handwritten text. Many applications involving number plate recognition, book scanning, and real time conversion of handwritten text benefit from OCR.

There are two types of OCR engine, handwritten and typewritten. Handwritten OCR engine recognize the characters written by the humans. On the other hand, typewritten OCR systems recognize the characters typed and scanned. Our approach is towards typewritten text, although this can be effectively updated for handwritten type.

One of the key features of OCR engine proposed is the simplicity. Approach for a better OCR engines are constantly made resulting with some improvements at every stage. However in this more complex modern world, the simplicity of the approach is being lost in hunt for better results. Unaware of many OCR engines using methodologies such as hidden Markov models, neural networks and natural language processing techniques, we devised a simple OCR system. This paper focuses on developing OCR engine using pixel and contour features and also uses mathematical parameters like intrinsic ratio. Our paper focuses on such a simple and effective S-OCR (Simple Optical Character Recognition) system.

II. S-OCR ALGORITHM

S-OCR algorithm focuses on recognition of digits using their individual characteristics feature. We have selected Arial Unicode MS font to showcase research results because this font produced some eccentric results as compared to synchronized results of other fonts.

We tested S-OCR algorithm on various font size and classified results based on four font size as will be indicated in subsequent analysis. Based on the large data set of varied font style and font size, our aim is to calculate the efficiency of S-OCR system.

The proposed S-OCR system follows three main phases:

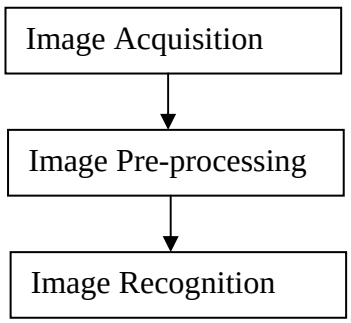

Fig. 1 three phases of S-OCR

A. Image acquisition

The proposed S-OCR starts with image acquisition process .It takes any typewritten image containing digits between 0-9 in either '.png' or '.jpg' format.

B. Image pre-processing

Following processes have been carried out in Image pre-processing.

1) *RGB to Gray Conversion*: The acquired image is converted from RGB channel to Gray for further pre-processing operations.

2) *Smoothing*: It simply removes the noise present in the image. It is also observed that performing adaptive thresholding on the image without the smoothing filter gives broken digits. After blurring, the digits obtained are clean and complete, which can be used for digit recognition.

3) *Adaptive Thresholding*: Once smoothed, the image is passed through an Adaptive thresholding algorithm. Applying window-wise thresholding helps in localizing the effect thus effectively bringing out the details which are perfect for our algorithm.

4) *Contour Detection:* Contour detection involves recognizing the boundary of digits and drawing a rectangle around that boundary. Contour detection is basically for determining the location and size of the digit. Example of contour detection is shown in Fig. 2.

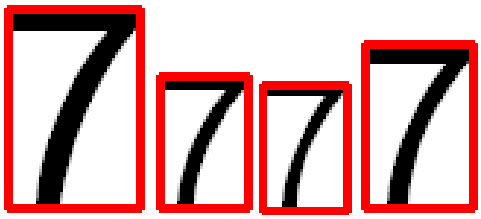

Fig. 2 contour detection

After the boundary is detected, the highlighted portion is cropped and is resized to 100x100 pixel size.

C. Image recognition

Every recognition algorithm has two steps:

- Determine features
- Classify (Runtime Recognition)

FEATURES:

1) Background Pixel Range: After resizing the cropped image to 100x100, we extracted the background pixels from the image, which are not filled with the data or the digit. As the background is uniform, it is easy to extract the background pixels. The background pixel range determines the amount of non-filled portion in the contour region.

Moreover it is beneficial to determine background pixel data rather than filled pixel data to gain accuracy. The background pixel data has been counted on the scale of 10,000**.**

2) Intrinsic Ratio: Intrinsic ratio is the ratio of Y-coordinate of centroid to the X-coordinate of centroid. The most important property of intrinsic coordinate is its stability. Modifications or glitches in the image cannot alter its intrinsic ratio.

The intrinsic ratio is calculated in following manner:

$$Cx = m[\text{`m01'}] / m[\text{`m00'}]$$
$$Cy = m[\text{`m01'}] / m[\text{`m00'}]$$

$$\text{Intrinsic Ratio } X = \text{int } ((Cy / Cx) * 100)$$

where m = Image Moments;

Cx = X-coordinate of centroid
Cy= Y-coordinate of centroid

Fig. 3 sample for centroid detection

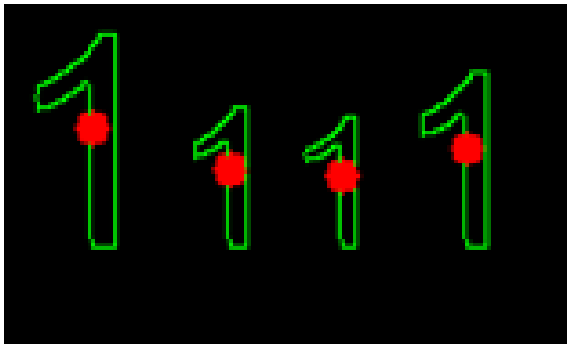

Fig. 4 centroid detection

3) Curved Contour Length: Curved contour length can be calculated by using number of curved coordinates present on the line. It picks one random coordinate in the filled pixels. As long as it finds the horizontal or vertical straight line joining the coordinate, it travels until the line gets a curve.

As long as it gets the curve, it picks that new coordinate and hence the number of coordinates keep increasing. Ideally a perfect rectangle should have contour length of four and a circle should have infinite contour length.

CLASSIFICATION:

The classification of the recognition is based on the characteristics feature. The classification is carried out in three steps:

1) Grouping using Background pixels Range: For grouping purpose, we collected background pixels for large quantity of data and derived the following range for grouping purpose.

TABLE I
RANGE DETERMINED BY BACKGROUND PIXELS

1

Digit	Font size range	Background pixels	Range (Out of 10000)
0	8 – 24	5654	5450 to 5950
	24 – 40	5498	
	40 – 56	5637	
	56 – 72	5907	
1	8 – 24	6182	5600 to 6200
	24 – 40	5813	
	40 – 56	5645	
	56 – 72	5842	
2	8 – 24	6071	5800 to 6100
	24 – 40	5835	
	40 – 56	5863	
	56 – 72	6024	
3	8 – 24	5906	5700 to 6250
	24 – 40	5722	
	40 – 56	5965	
	56 – 72	6201	
4	8 – 24	6332	5900 to 6350
	24 – 40	5944	
	40 – 56	6384	
	56 – 72	6175	
5	8 – 24	5766	5450 to 5800
	24 – 40	5487	
	40 – 56	5527	
	56 – 72	5667	
6	8 – 24	5081	4950 to 5400
	24 – 40	4988	
	40 – 56	5288	
	56 – 72	5371	
7	8 – 24	7285	6850 to 7300
	24 – 40	6890	
	40 – 56	6937	
	56 – 72	6906	
8	8 – 24	5109	4800 to 5400
	24 – 40	4847	
	40 – 56	5129	
	56 – 72	5351	
9	8 – 24	5269	5100 to 5450
	24 – 40	5126	
	40 – 56	5204	
	56 – 72	5401	

Based on the data derived from the extraction,digits can be classified in four non overlapping groups.

It is interesting to note that the range of 5 partially coincides with other digits. However in order to gain perfect accuracy, 5 is detected in the next step.

TABEL II .
DIGITS GROUP DETERMINED BY THEIR RANGE

Group	Digits in group	Range of groups
1	7	6850 to 7300
2	6, 8, 9	4800 to 5450
3	0, 1, 2, 3, 4	5450 to 6350
4	5	5450 to 5800

TABLE II clearly shows that the range of 7 does not coincide with any of the digits and can be easily distinguished from other digits. Hence 7 can be recognized at the first step of classification only. Other three groups are then sent to the next step for classification.

1) Intrinsic ratio: Rest groups are then further classified according to the intrinsic ratio range of each group.The achieved data for intrinsic ratio properties are shown in TABLE III.

TABLE III
RANGE OF INTRINSIC RATIO X FOR ALL DIGITS

Digit	Font size range	Cy	Cx	X = int((Cy/Cx)*100)	Range of X
0	8 – 24	49.51	57.65	86	82 to 86
	24 – 40	49.5	58.29	84	
	40 – 56	49.49	58.07	85	
	56 – 72	49.52	58.05	85	
1	8 – 24	47.05	88.30	53	50 to 55
	24 – 40	47.06	87.51	53	
	40 – 56	46.91	87.79	53	
	56 – 72	46.74	88.08	53	
2	8 – 24	45.02	63.49	70	67 to 72
	24 – 40	46.19	65.91	70	
	40 – 56	46.31	65.94	70	
	56 – 72	45.84	65.35	70	
3	8 – 24	49.89	67.00	74	73 to 77
	24 – 40	49.86	67.99	73	
	40 – 56	50.1	66.78	75	
	56 – 72	49.77	67.94	73	
4	8 – 24	48.00	65.28	73	69 to 75
	24 – 40	47.73	66.32	71	
	40 – 56	47.56	64.52	73	
	56 – 72	48.09	66.32	72	
5	8 – 24	52.75	60.85	86	84 to 86
	24 – 40	52.45	61.98	84	
	40 – 56	52.12	61.39	84	
	56 – 72	52.19	60.33	86	
6	8 – 24	50.44	56.09	89	87 to 89
	24 – 40	50.41	56.88	88	
	40 – 56	50.32	55.92	89	
	56 – 72	50.17	56.19	89	
8	8 – 24	49.77	59.02	84	82 to 84
	24 – 40	49.77	58.87	84	
	40 – 56	49.81	59.85	83	
	56 – 72	49.75	59.59	83	
9	8 – 24	48.56	62.09	78	76 to 78
	24 – 40	48.58	62.83	77	
	40 – 56	48.61	62.71	77	
	56 – 72	48.62	62.83	77	

Here it clearly shows that all three digits have different contour length range. Hence 2, 3 and 4 can be easily detected in this final step of classification.

The complete S-OCR algorithm is concluded in fig. 5.

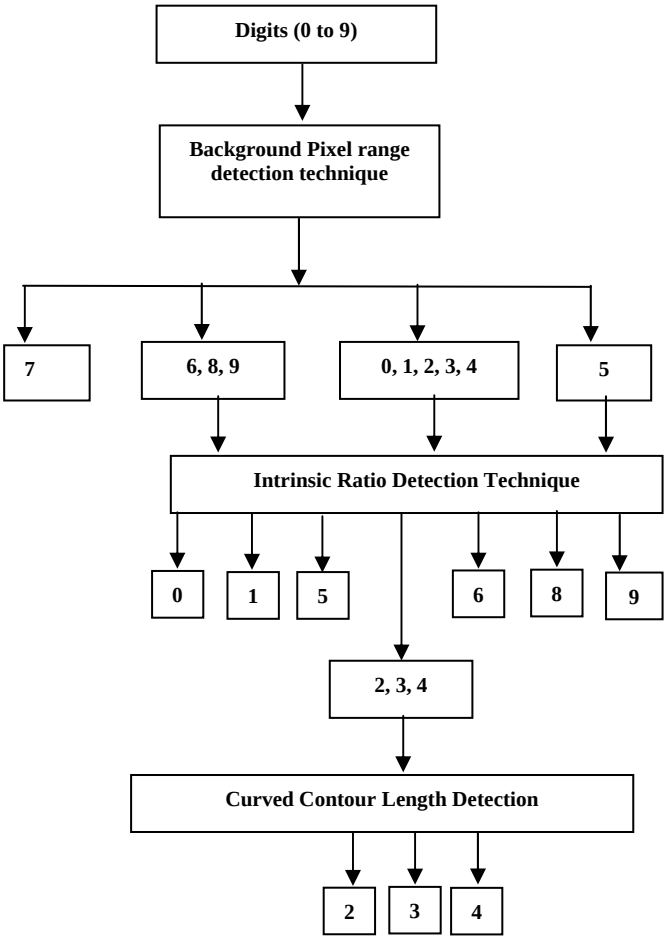

Fig. 5 S-OCR algorithm complete flow

III. Experimental Results

To illustrate the accuracy of the proposed S-OCR algorithm, the performance was tested on 100 typewritten images. The complete algorithm was implemented in Python (2.7.3) on a Dell Vostro (Intel Core2Duo @2.4GHz with 2.00 GB RAM).The recognition accuracy was 96.16%.

The example of the sample experimental image is shown in Fig. 6 and Fig. 7.The output for these experimental images is shown in Fig 8.

Fig. 6 Experimental Image – 1

Fig. 7 Experimental Image – 2

```
Python 2.7.3 (default, Apr 10 2012, 23:31:26) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ========================== RESTART ==========================
>>>
0
1
2
3
4
5
6
7
8
9
>>>
```

Fig. 8 Output of Experimental Images

IV. Conclusion and Future Work

In this paper, a Simple OCR engine for digits is proposed, the S-OCR algorithm uses the background pixel data range, intrinsic ratio range and curved contour length range for classification and recognition of digits with different sizes. It achieves around 96% accuracy.

Future work includes working with digits having different orientations and making the algorithm more robust and also extending the S-OCR algorithm to Alphabetic characters with various font size and font styles.

Acknowledgment

Authors are thankful to Abid Rehman for leading us towards proper direction. We thank Professor Pinkesh V. Patel and Professor Shital P. Thakkar for their lectures on various Image Processing techniques which we extensively made use of in this paper. We would like to extend special thanks to Professor Biren Patel for efforts in guiding us throughout other aspects of our paper.

References

[1] Ayatullah Faruk Mollah[1], Nabamita Majumder[2] , Subhadip Basu[3] and Mita Nasipuri[4] , "Design of an optical character recognition system for camera-based handheld devices " , IJCSI International Journal of Computer Science Issues , Vol. 8, Issue 4,No 1, July 2011

[2] Abdelwadood Mesleh[1], Ahmed Sharadqh, Jamil Al-Azzeh, MazenAbu-Zaher, Nawal Al-Zabin, Tasneem Jaber, Aroob Odeh and Myssa'a Hasn , "An optical character recognition" , Contemporary Engineering Sciences, Vol. 5, 2012, no. 11, 521 – 529

[3] Peter W. Frey, David J. Slate, "Letter recognition using Holland-Style adaptive classifiers", Machine Learning, 6, 161-182 (1991)

[4] http://www.opencv.org/documentation.html

[5] http://stackoverflow.com/questions/9413216/simple- digit-recognition-ocr-in-opencv-python

[6] https://github.com/abidrahmank/OpenCV2-Python/blob/master/Official_Tutorial_Python_Codes/3_imgproc/moments.py