IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-11, Nov 2018 E-ISSN: 2347-2693

Printed Numeral Recognition Using Structural and Skeleton Features

R. Vijaya Kumar Reddy[1*], Uppu Ravi Babu[2]

[1]Research Scholar, Dept. of. CSE, Acharya Nagarjuna University, Guntur, India
[2]Principal, DRK College of Engineering & Technology, Hyderabad, India

[*]*Corresponding Author:* vijayakumarr285@gmail.com

Available online at: www.ijcseonline.org

Accepted: 16/Nov/2018, Published: 30/Nov/2018

Abstract- In automatic numeral digit recognition system, feature collection is most important aspect for achieving high recognition performance. To attain this, we proposes model for printed numeral digit recognition using number of contours, skeleton features such as number of end points, number of horizental and vertical crossings Number of watersheds, and ratio between the number of foreground pixels in upper half-part and lower half-part of the numerical digit image. Based on these features the present study designed user defined classification algorithm for printed numerical digit recognition. To find the effectiveness of the proposed algorithm, these features are given as an input for standard classification algorithms like k–nearest neighbor classifier and other classification algorithms to evaluate the results. The experimental results prove that the proposed features are well suited for printed digit recognition for both user and standard classification algorithms. The novelty of the proposed method is size and shape invariant.

Keywords - Structural ,Skeleton Features.K-nn,Classification,Watersheds,contours.

I. INTRODUCTION

Deep Learning solutions have proved very successful on scanner-based or printed digit recognition [1], [2]. They have also shown good capability to handle complex inputs such as object recognition [2]. The scientific question that, when trying to address in this research is whether such deep learning systems are able to handle mixed contents, for example recognizing both text and printed digits inputs without separating them in two distinct problems. In a previous work, to address the problem of recognizing Sudoku puzzles from newspaper pictures taken with digital camera such as the ones embedded in our smart phones [3]. Analysis of documents and images with texts and digits/ numbers continues to be active research topics [4, 5, 6, 7, 8]. Hence the need for developing efficient digit and character recognition algorithms and techniques are still live in today. The feature extraction plays major role in numerical recognition system. Numbers of feature extraction methods are stated in the literature, like template matching, projection histogram, zoning, and various moment techniques [9] to enable for specific applications.

Based on the critical survey of existing methods, all the existing methods are used standard classification algorithms are used for classification. The present study aimed is to classify the isolated printed digits using user defined classification algorithm not with standard classification algorithms. The main objective of the present study is that classify the isolated printed digits with minimum number of features. To achieve this, the present approach proposed a new technique for classification without using any standard of isolated digits based on the feature extracts such as number of contours in an image [10], Number of end points of a thinned digit image, number of water reservoir sheds and ratio between the number of foreground pixels in upper half part and lower half-part of the numerical digit image Totally 5 features are extracted. Rest of the paper is organized as Section II contain the related work of Proposed system, database collection of printed numerals, Preprocessing and Features section, Section III contain the algorithm and architecture of the proposed system, Section IV contain the results and discussions and Section V contains Conclusions.

II. RELATED WORK

A. Proposed Method

The proposed method is primarily consists of 4 phases . In phase one, collection of the printed numerical data from various data sets. After collecting the numeral data preprocess the data i.e. exclusion of noise and renovation of gray level images into binary images and also the normalization of the binary images by using the normalization techniques in phase two. In the third phase, extracted the features from individual digit image and defined an algorithm for identification of printed numerals system in the final phase. The block diagram of the proposed method is shown below figure 1.

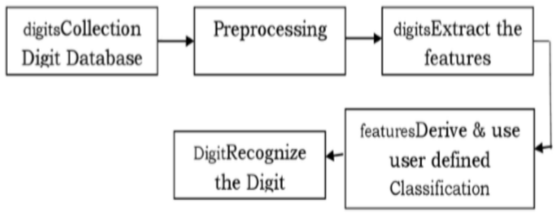


Figure 1: Block diagram proposed printed numerical digit identification system

B. Database Creation and Preprocessing

a) Printed numerical Image Database collection: No regular datasets are found for English digits. For recognition of printed numeral, the present study created own database. By using Ms-word option, typed the numerals of various font styles with different sizes. In each page 100 numerals are typed and take the print out. Totally, 150 pages are generated so that 15000 numerals are crated. A scanner which is flatbed is used for digitization, with images in gray tone at 300 dpi. Then this image stored with extension of BMP (Bit Map File) layout using a standard technique for converting them into black and white images. Data was physically extracted from scanned images and normalized into 50×50 size using a standard bi-cubic approach. After processing scanned images about digits and a total of 15000 (100×1150) images of numerals are obtained. The number of training and test samples is 5,000 and 10,000 respectively. The sample image of the created dataset is shown in figure 2.

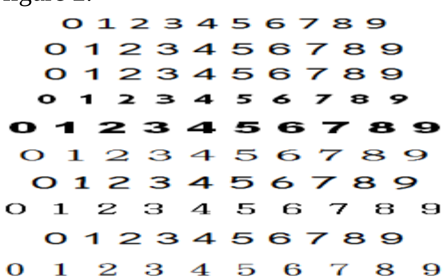


Figure 2: Printed Numerical database

b) *Pre Processing:* The identification of numerals can attain high performance based on pre-processing. The scanned digit database images used for this approach are in gray level images, convert this gray level images present in the database to binary images. Converting the gray level images into binary based on the threshold of intensity of pixel. After converting the images into binary, images may have remove elements one's (black) at unwanted places in the background image. For efficient classification, need to remove the surplus from undesirable places. The present study uses.

Digital images normal contains various types of noise. Noise present in it result to errors in the image acquisition process that result is effect to the values of true intensities of the real scene. There are numerous ways that noise can be introduced into an image, depending on how the image is created. For example:

- If the image is scanned from a photograph made on film, the film grain is a source of noise. Noise can also be the result of damage to the film, or be introduced by the scanner itself.
- If the image is acquired directly from camera the mechanism for gathering the data can introduce noise.
- Electronic broadcast of image data can introduce noise.

To simulate the effects of some of the problems listed above, the toolbox provides the imnoise function, which you can use to *add* various types of noise to an image. The examples in this section use this function.

➢ **Remove Noise by Linear Filtering:** By using the linear filtering to remove certain types of noise. Certain filters, such as averaging or Gaussian filters, are appropriate for this purpose

➢ **Remove Noise Using an Averaging Filter and a Median Filter:** To remove salt and pepper noise from an image using an averaging filter and a median filter to allow comparison of the results.

➢ **Remove Noise By Adaptive Filtering:** The Wiener filter tailors itself to the local image variance. Where the variance is large, this approach often produces better results than linear filtering. The adaptive filter is more selective than a comparable linear filter, preserving edges and other high-frequency parts of an image.

C. Feature Extraction

i. *The Number of Contours:* The number of contours is a structural feature. The contour is a margin of object pixel which separating object from a background. The contour contains the necessary information on the object shape. In computer vision, so many approaches are there to find the number of contours of a digit image. Some of the approaches are Freeman chain code approach [11], two-dimensional coding system [12], polygonal coding, and the connected component labeling algorithm [13] and etc. From them, the connected component labeling algorithm is most popular. In connected component labeling algorithm, the number of contours is equal to the number of background components (white components) minus one. For example, the digits 0, 2, 4, 6, 9 have two contours. The number of contours of the digits from 0 to 9 shown in table 1.

Table 1: Number of contours in Numerical digit

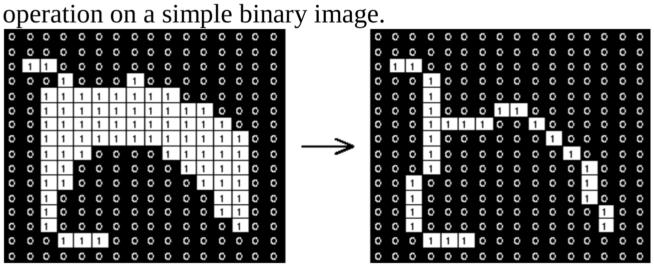
Digit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Number of contours	2	1	1	1	2	1	2	1	3	2

ii. Number of End Points: The end points of printed digit image can be extracted from the thinned / single pixel width image. **Thinning** is a morphological **operation** that is used to remove selected foreground pixels from binary **images**, somewhat like erosion or opening. It can be used for several applications, but is particularly useful for skeletonization and extract the skeleton features of a digit image.

The present study extract the 3 features from thinned image such as number of end points, number of horizental crossings and number of vertical crossings.

Thinning Algorithm

Thinning algorithm is a Morphological operation that is used to remove selected foreground pixels from binary images. It preserves the topology i.e. extent and connectivity of the original region while throwing away most of the original foreground pixels. Figure 3 shows the result of a thinning operation on a simple binary image.



(a) Original Image (b) Thinned Image

Figure 3: Thinning operation results

Thinning algorithms can be divided into two broad classes namely iterative and non-iterative. Although non-iterative algorithms can be faster than iterative algorithms they do not always produce accurate results.

Iterative Thinning

The Template-Based Mark-and-Delete Thinning Algorithms are very popular because of their reliability and effectiveness. This type of thinning processes uses templates, where a match of the template in the image, deletes the center pixel. They are iterative algorithms, which erodes the outer layers of pixel until no more layers can be removed [14]. Almost all iterative thinning algorithms use Mark-and-Delete templates including Stentiford Thinning Method. Both Stentiford and Zhang-Suen methods use Connectivity numbers to mark and delete pixels.

➢ **Connectivity Number:** The Connectivity number is to calculate of how many objects are coupled with a particular pixel. The following is the equation to compute Connectivity number.

$$C_n = \sum_{k \in S} N_k - (N_k \cdot N_{k+1} \cdot N_{k+2}) \tag{1}$$

Where: N_k is the color of the 8-neighbours of the pixel analyzed. N_0 is the center pixel. N_1 is the color value of the pixel to the right of the central pixel and the rest are numbered in counter clockwise order around the center.S = {1, 3, 5, 7}

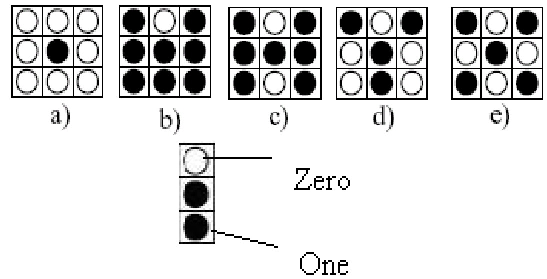


Figure 4: 3×3 windows used in thining operations

The following are the connectivity numbers for Figure 4 :a) Connectivity number = 0.b) Connectivity number = 1.c) Connectivity number = 2. d) Connectivity number = 3. e) Connectivity number = 4.

Stentiford Thinning Algorithm

It uses a set of four 3×3 templates to scan the image.

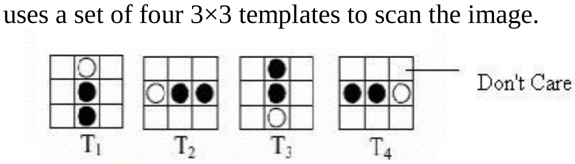


Figure 5: shows four templates used in thining operation.

The Stentiford Algorithm can be defined as following:

1. Find a pixel position (i, j) where the pixels in the image equal to those in template T1. With this template all pixels along the top of the image are removed moving from left to right and from top to bottom.
2. If the central pixel *is not an **endpoint**, and has connectivity number = 1*, then mark this pixel for deletion.

Endpoint pixel: A pixel is considered an end point if it is link to just one other pixel. That is, if a black pixel has only one black neighbour out of the eight possible neighbours.

3. Repeat steps 1 and 2 for all pixel locations matching T1.
4. Repeat steps 1 to 3 for the remaining templates: T2, T3, and T4. T2 will match pixels on the left side of the object, moving from bottom to top and from left to right. T3 will select pixels along the bottom of the image and move from right to left and from bottom to top. T4 locates pixels on the right side of the object, moving from top to bottom and right to left.
5. Set to white the pixels marked for deletion.

Figure 6: some examples of the thinning process using the Stentiford Algorithm

After thinning operation is performed find the number of end points of each digit.

End point (EP) [15] is a digit point (black pixel) with only one black neighbor. The patterns used to find endpoints of an image when the image is scan from top left corner to bottom right corner in shown in figure 7. The table 2 shows the number of end points of the each numerical digit.

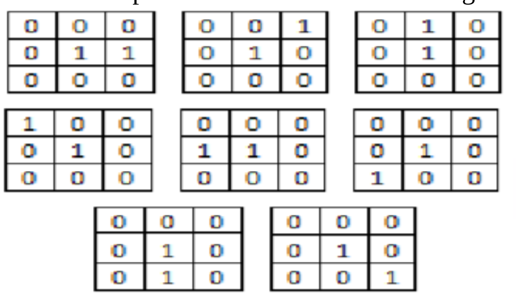


Figure 7: End point pattern templates

Table 2: Number of End points of in Numerical digit

Digit	0	1	2	3	4	5	6	7	8	9
Number of end points	0	2	2	3	2	2	1	2	0	1

iii.Number of horizental/ vertical crossings:

Vertical/ Horizontal crossings are determined by adding the number of white-black-white transfers when scanning the image's intensity values on a vertical/ horizontal line. The line is passes through the center of mass of the digit image. The template for the vertical/ horizontal crossing is shown in figure 8. The number of horizental and horizental cross of the digits are shown in figure 9.

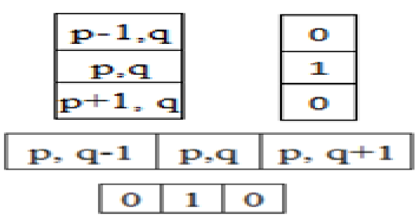


Figure 8: Template for vertical/horizontal crossing and their values in that position

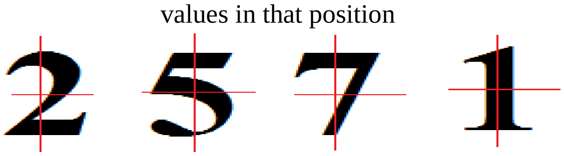


Figure 9: horizental and vertical of the the digit 2, 5, 7 and 1

iv.Number of reservoirs

The water reservoir principle is as follows. If water is poured from a side of a component, the cavity regions of the component where water will be stored are considered as reservoirs [12]. The cavity regions of the components where water will be stored are considered as reservoir. The printed digits generate reservoirs which are used for recognition. Now, the present study discusses the water reservoir extraction scheme of printed Digits.

Top reservoir: The top reservoirs of a digit mean that the reservoirs obtained when water is poured from top of the digit.

Bottom reservoir: The bottom reservoirs of a digit means that the reservoirs obtained when water is poured from bottom of the digit. **Left (right) reservoir:** If water is poured from the left (right) side of a component, the cavity regions of the digit where water will be stored are considered as left (right) reservoirs. The figure 10 illustrates the reservoirs. Number of sides water will preserve in a digit is considering as one of the feature for recognition. For Recognizing the printed digits 1,2,5 and 7 use this feature. Digit 1 has one side, the digits 2, 5, and 7 have two sides.

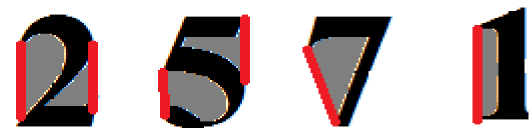


Figure 10 : number of water reservoirs of digits 1, 2, 5 and 7

v.Ratio Of Number of Pixels In Upper Half to Lower Half Part

The number foreground pixel is in top half part the digit to bottom half part of the digit image is considered as feature for recognition of digits 6 and 9.Width of top line of the Digit: For classifying it is another important feature for classifying the digits 2 and 5. The digit '2' has less width compared to 5. The figure 11 shows the difference of the width of the digit 2 and 5.

Figure 11: width of the top line of the digit.

III. METHODOLOGY

Proposed User Defined Algorithm:

Based on the specified features extracted from the printed digit, the following algorithm is designed for the recognition of printed numerals are shown in algorithm1.

Algorithm1: Recognition of Printed Digits

Input: Isolated Printed numeral image

Output: Classification of the Numeral Digit 0 or 1 or 2 or 3 or 4 or 5 or 6 or 7 or 8 or 9.

Method: Structural, Statistical features extraction.

Step 1: Pre Process the image

Step 2: Find the Number of Contours (NoC) and classify the digit as Group1 or Group2 or digit 8.

Step 3: Find out the Skeleton of input image and calculate the Skeleton feature number of end points(NoEP) in Group1 and Group2.

Step4: Based on NoEP in digits in Group1 further divide into 1 sub-groups Group1a, and classify the digit 3.
Step5: Calculate the Number of water sheds (NoW) of the Digits in Group1a.
Step6: Based on NoW values further divide into Subgroup Group1aa and Group1ab.
Step7: Calculate the Number of vertical crossings (NoVC) of the digits in Group1aa.
Step8: Based on the NoVC values classify the image is either digit 1 or 7.
Step9: Calculate the number of pixels in top line and bottom line of the digits in Group1ab.
Step10: Based on the ratio, classify the digit either 2 or 5.
Step 11: Based on NoEP in digits in Group2 further divide into 1 sub-groups Group2a, and classify the digit 0 and 2.
Step12: Calculate the ratio between the numbers of foreground pixels in upper half part and lower half-part (ULR) of the numerical digit images in Grop2a.
Step13: Based on the ULR values classify the digit either 6 or 9.

The graphical representation of the user define algorithm is shown in figure 12.

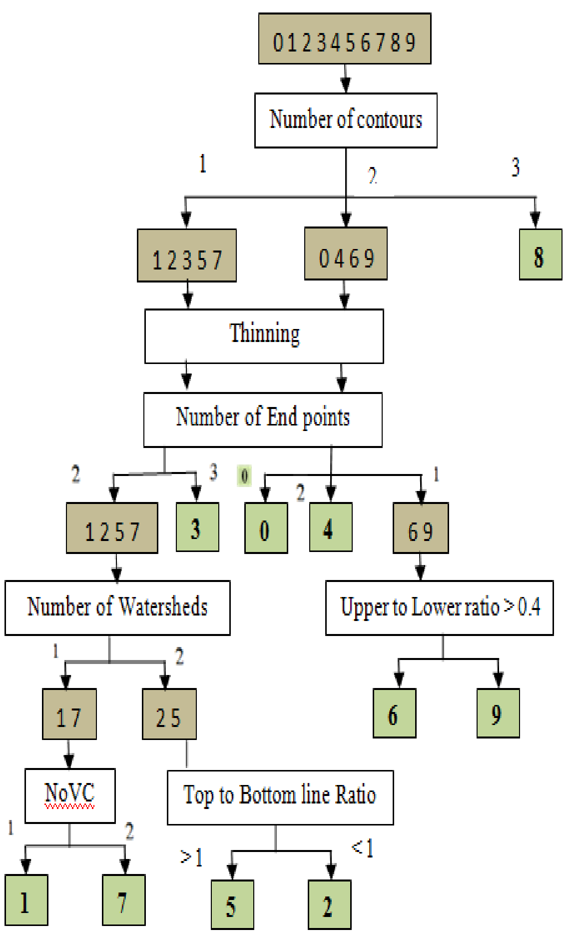


Figure 12: Graphical representation of user defined algorithm

From the above algorithm, it is identified that the proposed method needs the feature at maximum 4 for the classification of handwritten digits every digit does not need all these 4 features. The number of features required to recognize the digits is shown in the table 3.

Table.3: Feature count required for each digit

Digit	0	1	2	3	4	5	6	7	8	9
Number of Features	2	4	4	2	3	4	3	4	1	3

IV. RESULTS AND DISCUSSION

The present experiments is done on a PC machine with i3 processor 2.7GHz CPU and 2 GB RAM memory under MatLab 10.0 platform. From the training set images extract the feature values which are specified in above section and stored in Feature Vector(FV) and extract the features of test database and stored in Test Vector (TV).

Classifier is depends greatly on the characteristics of the data to be classified. There is no particular classifier that works best for all given problems. Various practical tests have been done to measure classifiers performance and to find the features of data that determine classifier performance. Determining a appropriate classifier for a given problem is however still more an art in research. An fascinating problem in pattern recognition yet to be solved is the relationship between the problem to be solved (data to be classified) and the performance of various pattern recognition algorithms (classifiers). In this we used user-defined classification algorithm and k-Nearest-Neighbor Classifier classify test database digit images.

A. *User defined algorithm*: By using the algorithm defined in section II, the digits are classified. The proposed algorithm is tested with 15000 images and the individual result of the test database is listed in table 4.

Table 4: Results of Numerical Recognition for 15000 test images using use defined classifier

Digit	No. of images	Correctly Classified	Not Correctly Classified	% Accuracy
0	1500	1500	0	100.00
1	1500	1500	0	100.00
2	1500	1500	0	100.00
3	1500	1500	0	100.00
4	1500	1500	0	100.00
5	1500	1500	0	100.00

6	1500	1500	0	100.00
7	1500	1500	0	100.00
8	1500	1500	0	100.00
9	1500	1500	0	100.00
Average Recognition Percentage				100.00

B. K-Nearest neighbor classification (k-nn Classifier:
The proposed method uses k-nearest neighbor (k-*nn*) classification algorithm for classifying the test database digit images in test database using the feature vector of training database.. In K-*nn* object is classified to a particular class which has bulk of votes. In the k-*nn* classification, compute the distance between feature values of the test sample and the feature vector values of every training image and the class of greater part among the k-nearest training samples is based on the Euclidian distance measures. The training vector is a multidimensional array. Each row in an array contains feature values and corresponding class label of the training images whereas test vector contains only feature values. In classification process, for each row in test vector assign the class label based on the Euclidian distance measures and number of neighbors (k) considered. The k value is defined by the user. the k value used in the present study is 3.

Table 5: Results of Numerical Recognition for 15000 test images using k-*nn* classifier

Digit	No. of images	Correctly Classified	Not Correctly Classified	% Accuracy
0	1500	1500	0	100.00
1	1500	1498	2	99.87
2	1500	1497	3	99.80
3	1500	1495	5	99.67
4	1500	1496	4	99.73
5	1500	1497	3	99.80
6	1500	1498	2	99.87
7	1500	1496	4	99.73
8	1500	1500	0	100.00
9	1500	1498	2	99.87
Average Recognition Percentage				99.83

C. *Analysis of the Proposed Method:*

To analyze the proficiency of the proposed system, the outcomes of the proposed method are analyzed in gotten Ten Cycle Cross Validation (TCCV) approach. Ten Cycle Cross Validation (TCCV) approach: In TCCV approach results analysis strategy, the entire digit data base i.e. 15000 are divided into 10 sets. Each set consists of 1500 digit images. Every set must contain ten classes (0-9) of digit images. In TCCV approach results are analyzed in 10 Cycles. In cycle 1, first set is treated as a sample set and remaining 9 sets are taken as a test dataset. The % of digit grouping of the proposed strategy in ten cycles are listed in tables 6 to 15 individually.

Table 6: % of recognition of the proposed method in cycle-1 of TCCV approach

Digit	No. of images	Correctly Classified	Not Correctly Classified	% Accuracy
0	1350	1350	0	100.00
1	1350	1348	2	99.85
2	1350	1349	1	99.93
3	1350	1348	2	99.85
4	1350	1349	1	99.93
5	1350	1350	0	100.00
6	1350	1349	1	99.93
7	1350	1348	2	99.85
8	1350	1350	0	100.00
9	1350	1349	1	99.93
Average Recognition Percentage				99.93

Table 7: % of recognition of the proposed method in cycle-2 of TCCV approach

Digit	No. of images	Correctly Classified	Not Correctly Classified	% Accuracy
0	1350	1350	0	100.00
1	1350	1349	1	99.93
2	1350	1348	2	99.85
3	1350	1349	1	99.93
4	1350	1349	1	99.93
5	1350	1350	0	100.00
6	1350	1349	1	99.93
7	1350	1349	1	99.93
8	1350	1350	0	100.00

9	1350	1349	1	99.93
Average Recognition Percentage				99.94

Table 8: % of recognition of the proposed method in cycle-3 of TCCV approach

Digit	No. of images	Correctly Classified	Not Correctly Classified	% Accuracy
0	1350	1350	0	100.00
1	1350	1349	1	99.93
2	1350	1349	1	99.93
3	1350	1349	1	99.93
4	1350	1348	2	99.85
5	1350	1350	0	100.00
6	1350	1349	1	99.93
7	1350	1348	2	99.85
8	1350	1350	0	100.00
9	1350	1349	1	99.93
Average Recognition Percentage				99.93

Table 9: % of recognition of the proposed method in cycle-4 of TCCV approach

Digit	No. of images	Correctly Classified	Not Correctly Classified	% Accuracy
0	1350	1349	1	99.93
1	1350	1350	0	100.00
2	1350	1349	1	99.93
3	1350	1350	0	100.00
4	1350	1348	2	99.85
5	1350	1350	0	100.00
6	1350	1349	1	99.93
7	1350	1350	0	100.00
8	1350	1350	0	100.00
9	1350	1349	1	99.93
Average Recognition Percentage				99.96

Table 10: % of recognition of the proposed method in cycle-5 of TCCV approach

Digit	No. of images	Correctly Classified	Not Correctly Classified	% Accuracy
0	1350	1350	0	100.00
1	1350	1349	1	99.93
2	1350	1348	2	99.85
3	1350	1349	1	99.93
4	1350	1350	0	100.00
5	1350	1349	1	99.93
6	1350	1350	0	100.00
7	1350	1349	1	99.93
8	1350	1349	1	99.93
9	1350	1350	0	100.00
Average Recognition Percentage				99.95

Table 11: % of recognition of the proposed method in cycle-6 of TCCV approach

Digit	No. of images	Correctly Classified	Not Correctly Classified	% Accuracy
0	1350	1349	1	99.93
1	1350	1350	0	100.00
2	1350	1350	0	100.00
3	1350	1348	2	99.85
4	1350	1350	0	100.00
5	1350	1349	1	99.93
6	1350	1350	0	100.00
7	1350	1349	1	99.93
8	1350	1350	0	100.00
9	1350	1350	0	100.00
Average Recognition Percentage				99.96

Table 12: % of recognition of the proposed method in cycle-7 of TCCV approach

Digit	No. of images	Correctly Classified	Not Correctly Classified	% Accuracy
0	1350	1350	0	100.00

1	1350	1349	1	99.93
2	1350	1350	0	100.00
3	1350	1348	2	99.85
4	1350	1350	0	100.00
5	1350	1349	1	99.93
6	1350	1349	1	99.93
7	1350	1349	1	99.93
8	1350	1350	0	100.00
9	1350	1349	1	99.93
Average Recognition Percentage				99.95

Table 13: % of recognition of the proposed method in cycle-8 of TCCV approach

Digit	No. of images	Correctly Classified	Not Correctly Classified	% Accuracy
0	1350	1349	1	99.93
1	1350	1350	0	100.00
2	1350	1349	1	99.93
3	1350	1349	1	99.93
4	1350	1349	1	99.93
5	1350	1350	0	100.00
6	1350	1350	0	100.00
7	1350	1350	0	100.00
8	1350	1350	0	100.00
9	1350	1350	0	100.00
Average Recognition Percentage				99.97

Table 14: % of recognition of the proposed method in cycle-9 of TCCV approach

Digit	No. of images	Correctly Classified	Not Correctly Classified	% Accuracy
0	1350	1350	0	100.00
1	1350	1350	0	100.00
2	1350	1348	2	99.85
3	1350	1348	2	99.85
4	1350	1349	1	99.93
5	1350	1350	0	100.00
6	1350	1349	1	99.93
7	1350	1350	0	100.00
8	1350	1350	0	100.00
9	1350	1349	1	99.93
Average Recognition Percentage				99.95

Table 15: % of recognition of the proposed method in cycle-10 of TCCV approach

Digit	No. of images	Correctly Classified	Not Correctly Classified	% Accuracy
0	1350	1350	0	100.00
1	1350	1349	1	99.93
2	1350	1349	1	99.93
3	1350	1349	1	99.93
4	1350	1348	2	99.85
5	1350	1350	0	100.00
6	1350	1349	1	99.93
7	1350	1350	0	100.00
8	1350	1349	1	99.93
9	1350	1348	2	99.85
Average Recognition Percentage				99.93

V. CONCLUSION

To achieve this, we proposes model for printed numeral recognition using number of contours, skeleton features such as number of end points, number of horizental and vertical crossings Number of watersheds, and ratio between the number of foreground pixels in upper half part and lower half-part of the numerical digit image. The novelty of this method is that, it is free from size normalization. The present approach defined a classification of isolated printed digits with good classification results. The proposed method was tested with large database. The proposed approach extracts only 5 features. The result obtained in this encourages me to develop an approach for classification of handwritten digit.

REFERENCES

[1] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural Comput., vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[2] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in Proceedings of the 26th Annual International Conference on Machine Learning, ser. ICML '09. New York, NY, USA: ACM, 2009, pp. 609–616.

[3] B. Wicht and J. Hennebert, "Camera-based sudoku recognition with deep belief network," in Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of, Aug 2014, pp. 83– 88.

[4] Chen, G., Li, Y. and Srihari, S.N. (2016) Word Recognition with Deep ConditionalRandom Fields. Proceedings of the IEEE International Conference on Image Processing(ICIP), Phoenix, AZ, 25-28 September 2016, 1928-1932.

[5] arulla, G.A., Murru, N. and Rossini, R. (2016) A Fuzzy Approach for Segmentationof Touching Characters. arXiv:1612.04862v1. https://arxiv.org/abs/1612.04862

[6] Jaderberg, M., Simonyan, K., Vedaldi, A. and Zisserman, A. (2014) Reading Text inthe Wild with Convolutional Neural Networks. arXiv preprint arXiv:1412.1842

[7] Liu, Z., Li, Y., Ren, F. and Yu, H. (2016) A Binary Convolutional Encoder-DecoderNetwork for Real-Time Natural Scene Text Processing. arXiv:1612.03630v1.https://arxiv.org/abs/1612.03630v1

[8] Yin, X.C., Yin, X., Huang, K. and Hao, H.W. (2014) Robust Text Detection in NaturalScene Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36, 970-983. https://doi.org/10.1109/TPAMI.2013.182

[9] Oivind Trier, Anil Jain, Torfiinn Taxt, ―A feature extraction method for character recognition-A survey pattern Recg, vol 29, No 4, pp641-662, 1996

[10] P.Nagabhushan, S.A.Angadi, B.S.Anami, ―A fuzzy statistical approach of Kannada Vowel Recognition based on Invariant Moments‖, Proc. Of NCDAR-2003, Mandy, Karnataka, India, pp275-285, 2003

[11] G. G. Rajput, Rajeswari Horakeri, Sidramappa Chandrakant "Printed and Handwritten Kannada NumeralRecognition Using Crack Codes and Fourier Descriptors Plate" RTIPPR, 2010. IJCA Special Issue on "RecentTrends in Image Processing and Pattern Recognition" pp.53-58

[12] Graham, D.N "Image transmission by two-dimensional contour coding" Proceedings of the IEEE Volume: 55,Issue: 3 Pages: March 1967 336 – 346

[13] U. Pal, T. Wakabayashi, N. Sharma and F. Kimura,"Handwritten Numeral Recognition of Six Popular IndianScripts", In Proc. 9th International Conference on Document Analysis and Recognition. pp. 749-753, Curitiba,Brazil, September 24-26, 2007

[14] Seong- W han Lee, "Multi layer cluster neural network for totally unconstrained handwriritten numeralrecognition. Neural Networks. Vo I. 8. No 5. pp.409-4 1 8, 1984

[15] Image Processing Techniques For Machine Vision [online].Available:http://www.eng.fiu.edu/me/robotics/elib/am_st_fiu_ppr_2000.pdf

Authors Profile

R. Vijaya Kumar Reddy Research Scholar, Dept. of. CSE, Acharya Nagarjuna University, Guntur. He has published more than 10 research paper in International journals. He as 8 years of teaching experience and 4 years of Research Experience. He is a life member of ISSE, IAENG and ISTE.

Dr. U Ravi Babu obtained his Ph.D in computer science engineering from Acharya Nagarjuna University. He completed MSc Information Systems (IS) from AKRG PG College, Andhra University in the year 2003 and M. Tech in Computer Science & Engg from RVD University in the year 2005. He is a member of SRRFGIET, Rajahmundry. He has published 30 research papers in various Inter National journals and 10 papers in various National and International conference proceedings. He is working as a Principal, DRK College of Engineering & Technology Hyderabad from 2017 to till date. He is a life member of ISCA, IAENG and IRS.