Advances in Handwriting Recognition: Using Python

Ayush Kumar
Ayush_kumar7.scsebtech@galgotiasuniversity.edu.in
Galgotias University
Uttar Pradesh, INDIA

Avinash Kumar Singh
Avinash_kumar..scsebtech@galgotiasuniversity.edu.in
Galgotias University
Uttar Pradesh, INDIA

Abstract— This paper proposes a recognition model for English handwritten (lowercase, uppercase and letter) character recognition that uses Freeman chain code (FCC) as the representation technique of an image character. Chain code representation gives the boundary of a character image in which the codes represent the direction of where is the location of the next pixel. An FCC method that uses 8-neighbourhood that starts from direction labelled as 1 to 8 is used. Randomized algorithm is used to generate the FCC. After that, features vector is built. The criteria of features to input the classification is the chain code that converted to 64 features. Support vector machine (SVM) is chosen for the classification step. NIST Databases are used as the data in the experiment. Our test results show that by applying the proposed model, we reached a relatively high accuracy for the problem of English handwritten recognition.

Keywords- Freeman chain code (FCC), Heuristic method, randomized algorithm, features vector, Support vector machine (SVM).

I. INTRODUCTION

Handwritten character recognition (HCR) is a part of off-line character recognition. Handwritten characters have infinite variety of style from one person to another person. Due to this wide range of variability, it is difficult to recognize by a machine.

Although the research in Optical Character Recognition (OCR) has been going on for last few decades, the goal of this area is still out of reach. Most of the researchers have tried to solve the problems based on the image processing and pattern recognition techniques. The result of this research is the accumulation of many algorithms for classification using the rough representation-in pixels-of the character or feature vector representation [1].

Normally, HCR can be divided into three steps namely pre-processing, feature extraction and classification. Pre-processing stage is to produce a clean character image, it is can be used directly and efficiently by the feature extraction stage. Feature extraction stage is to remove redundancy from data. And lastly, classification stage is to recognize characters or words.

Feature extraction in HCR is a very important field of image processing and object recognition. Fundamental component of characters called features. As in many practical problems, it is often not easy to find those with most effective features [2].

Types of features are depended on the system in which they are implemented. One of them is shape feature. There are two procedures for extracting these features. The first procedure is boundary extraction that based on the outer boundary shape. The second procedure is region extraction that works with the whole shape as an object.

In this paper, heuristic character recognition is explored using a randomized algorithm. The proposed method is used to minimize the length of FCC from a thinned binary image (TBI). The main problem in representing the characters using FCC is the length of the FCC that depends on the starting points. Also, during FCC generation that require traversing each pixel (or node) of the character, it is often to find the problem of finding several branches and revisiting the same nodes. To solve this problem, heuristic is used to generate the FCC correctly to represent the characters. The classification stage must correctly represent and distinguish each character. SVM classifier is used to recognize the characters.

The rest of this paper has the following structure. Section 2 presents the architecture of the recognition model which are pre-processing, specific extraction methods, and SVM classifier. Section 3 presents the experimental results of English character (lowercase and uppercase letter) recognition using NIST (National Institute of Standards and Technology) Databases and followed by conclusion in section 4.

II. THE ARCHITECTURE MODEL

This section describes about the architecture model of HCR in detail in “Fig. 1”.

A. *Pre-processing*

Pre-processing stage involves all of the operations to produce a clean character image, so that it can be used directly and efficiently by the feature extraction stage. Before extracting the features from an image, a sequence of simple, common pre-processing is applied in order to standardize the data and make it feasible to the recognition algorithms and to reduce complexity [3].

This pre-processing stage only involves of a thinning process. Thinning is an important pre-processing step in OCR. The purpose of thinning is to delete redundant information and at the same time retain the characteristic features of the image. Thinning is applied to find a skeleton of a character. Skeleton is an output of thinning process. The resulting skeleton images of the characters are shown in "Fig. 2".

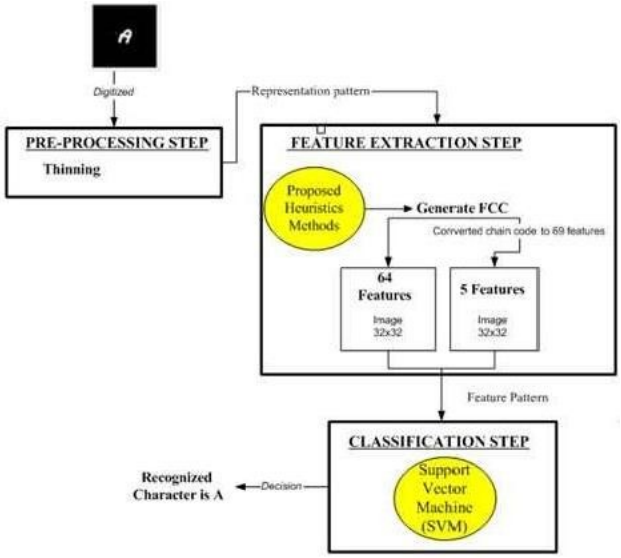

Figure 1. The architecture of recognition system

Figure 2. Skeleton produced by thinning process

B. *Feature Extraction*

The main objective of feature extraction is to remove redundancy from data. The task of human expert is to select features that allow effective and efficient recognition of pattern. Feature extraction is a very important in recognition system because it is used by the classifier to classify the data.

Chain code is one of the representation techniques that is useful for image processing, shape analysis and pattern recognition fields. Chain code representation gives the boundary of character image in which the codes represent the direction of where is the location of the next pixel. The first approach of chain code was introduced by Freeman in 1961 that is known as Freeman Chain Code (FCC) [8]. There are two directions of chain code, namely 4-neighborhood and 8-neighborhood as shown "Fig. 3". As in many papers usually the researchers start from zero until seven in 8-neighbourhood. But for this paper will start from one until eight because is easy for distinguish direction or non direction (value is zero) of chain code.

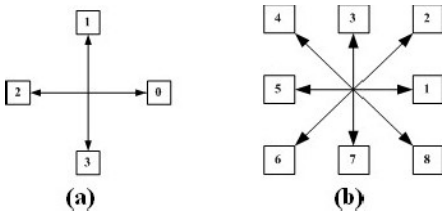

Figure 3. Freeman chain codes : (a) 4-Neighborhood and (b) 8-Neighborhood

Heuristic is a method to find a solution that is closed to the best but it does not guarantee that the best will be found. In this paper a heuristic methods are proposed which is randomized algorithm.

The pseudo-code of randomized algorithm is depicted in Table 1 and the description of randomized algorithm is shown in "Fig. 4".

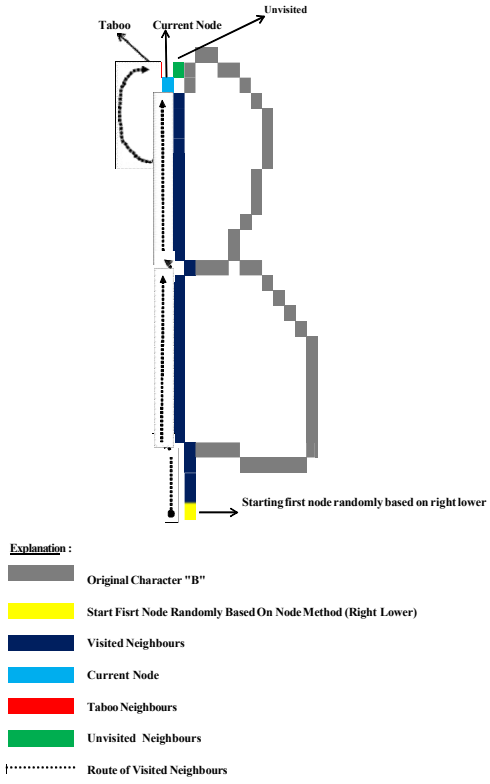

Figure 4. Description of randomized algorithm

TABLE I. THE PSEUDO-CODE OF RANDOMIZED ALGORITHM

```
Initialize Data
while Termination Not Met do
   Select First Node Randomly
   {Node-Method, End-Node-Method}
        while Number of Visited Node < Number of Node do
              if there are Unvisited neighbours
                    Select One Node Randomly
              elseif there are Visited neighbours
                    Select One Node Randomly
              elseif there are Taboo neighbours
                    Select One Node Randomly
              endif
        endwhile
endwhile
Display Solutions
```

The procedure is as following: start from the first node, which is node-method and end-node-method. Node- method is to find the first character for every aspects boundary such as left upper, left lower, right upper and right lower. End-node-method is to find the first character based on the end position of a character.

In this randomized algorithm, if the number of visited node less than the number of nodes, there would be three kinds of characteristics, which are unvisited, visited and taboo neighbours. Unvisited neighbours are nodes that never went through the route searching. Visited neighbours indicate the nodes that have went through the route searching. Taboo neighbours are used to keep track of the visited search space and revisited node with one step after current node.

The procedure is as following: start from the first node, which is node-method and end-node-method. Node- method is to find the first character for every aspects boundary such as left upper, left lower, right upper and right lower. Endnode-method is to find the first character based on the end position of a character.

The criteria of features to input the classification stage is the chain code that converted to features that became 69 features (8 directions of chain code x 8 routes of FCC = 64 features + 5 extra features = 69 features). Sixty four features are created from the generated FCC and five extra features are from the calculated values of ratio-upper, ratio-right, ratio-height-weight, ratio-height and number of string character. The ratio-upper is calculated from firstly by cropping the image and then defining the centre of the image character. After that, the total number of upper character is divided with the total number of character. This is done similarly to the ratio-right, ratio-height-weight and ratio-height. The formula of height character as shown in Equation 1 below:

$$Height = \frac{height\ of\ character}{height\ of\ image\ (number\ of\ pixel\ image)}$$

$$Ratio = \frac{frequency}{total} \quad (2)$$

C. *Building SVM Classifier*

The concept of SVM (Support Vector Machine) was introduced by Vapnik and co-workers [4]. It gains popularity because it offers the attractive features and powerful machinery to tackle the problem of classification i.e., we need to know which belongs to which group and promising empirical performance.

The SVM is based on statistical learning theory. SVM's better generalization performance is based on the principle of Structural Risk Minimization (SRM) [4]. The concept of SRM is to maximize the margin of class separation. The SVM was defined for two-class p oblem and it looked for optimal hyper-plane, which maximized the distance, the margin, between the nearest examples of both classes, named SVM [5]. For information detail about SVM can be seen in [4,5,6].

We have utilised radial basis function for its kernel function. The input feature sets were the directional features (169-dimension). All the SVM' are trained with the respective training feature sets and the results explored by using separate test data which are lowercase and uppercase letters.

III. EXPERIMENTAL RESULTS

In this study, we used NIST databases as experiment data. This section describes the training and testing sets, and the experimental results.

A. *Data Set*

For experimental analysis, we considered 189,411 samples for lowercase letters, 217,812 for uppercase letters and 407,223 for the combination of uppercase and lowercase letters.

B. *Performance of the Proposed System*

We built three datasets serving for training and testing:

1. TrainData (TD) 1: lowercase with 189,411 samplesWhich are 151,533 as training and 37,878 as testing.
2. TrainData (TD) 2: uppercase with 217,812 samples
3. which are 169,099 as training and 48,713 as testing.
4. TrainData (TD) 3: lowercase and uppercase letter with 407,223 samples which is 320,632 as training and 86,591 as testing.

Normalisation is performed after obtaining the chain code for all characters. The purpose of normalisation is to distinguish the differences in size of image characters. During normalisation it is important to find the ratio of every character. The formula of ratio as shown in Equation 2 below:

We built three datasets that serve for recognition. The data samples are divided into 20% for testing and 80% for training. Radial basis function is used to set the SVM model. Finally, the relationships between the kernel function parameters are obtained for the accuracy of the testing and training. The test results from data sets of English handwritten characters are shown in Table 2.

From the results depicted in Table 2, it can be seen that the proposed method have successfully recognised the handwriting characters from NIST database with high accuracy of more than 86% for the first dataset, 88% for second data set and 73% for third dataset.

C. *Erroneous Samples*

From the experiment, not all image characters in the NIST database can be used in the experiment. For instance, NIST database has 190,998 lowercase letters; however, only 189,411 samples can be used due to the very poor quality samples and sometimes broken parts, which made recognition task more difficult.

IV. CONCLUSION

This paper proposes a model for handwritten English handwritten character recognition. The proposed model starts with pre-processing. The pre-processing stage involves all of the operations to produce a clean character image, so that it is can be used directly and efficiently by the feature extraction stage. Thinning algorithm is used in the pre-processing stage that produced a skeleton of a character. The second step is feature extraction. FCC is generated from the characters that used as the features for classification. The main problem in representing the characters using FCC is the length of the FCC that depends on the starting points. Also, during FCC generation that require traversing each pixel (or node) of the character, it is often to find the problem of finding several branches and revisiting the same nodes. To solve this problem, heuristic is used to generate the FCC correctly to represent the characters. The third step is classification using the features generated from FCC. Our recognition model was built from SVM classifiers. Our test results shows that applying the proposed model, we reached a relatively high accuracy for the problem of English handwritten recognition.

V. ACKNOWLEDGMENT

The authors honourably appreciate Ministry of Science, technology and Innovation (MOSTI) for the ScienceFund grant with vot number 79369 and Research Management Centre (RMC), University of Technology Malaysia (UTM) for the support in making this projects success.

7. References

[1] Vapnik, V.N., 2019. An overview of statistical learning theory. IEEE transactions on neural networks, 10(5), pp.988-999.

[2] Mantas, J., 2015. An overview of character recognition methodologies. Pattern recognition, 19(6), pp.425-430.

[3] Roberts, G.I., and Samuels, M.T., 2018. Handwriting remediation: A comparison of computerbased and traditional approaches. The Journal of Educational Research, 87(2), pp.118-125.

[4] Chang, C.J., Lo, C.O. and Chuang, S.C., 2020. Applying Video Modeling to Promote the Handwriting Accuracy of Students with Low Vision Using Mobile Technology. Journal of Visual Impairment & Blindness, 114(5), pp.406-420.

[5] Potanin, M., Dimitrov, D., Shonenkov, A., Bataev, V., Karachev, D. and Novopoltsev, M., 2021. Digital Peter: Dataset, Competition and Handwriting Recognition Methods. arXiv preprint arXiv:2103.09354.

[6] [Bahlmann, C., Haasdonk, B. and Burkhardt, H., 2012, August. Online handwriting recognition with support vector machines-a kernel approach. In Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition (pp. 49-54). IEEE.]

[7] Zanchettin, C., Bezerra, B.L.D. and Azevedo, W.W., 2020, June. A KNN-SVM hybrid model for cursive handwriting recognition. In The 2012 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[8] Fanany, M.I., 2019, May. Handwriting recognition on form document using convolutional neural network and support vector machines (CNN-SVM). In 2017 5th international conference on information and communication technology (ICoIC7) (pp. 1-6). IEEE.

[9] Kaur, R.P., Jindal, M.K. and Kumar, M., 2021. Text and graphics segmentation of newspapers printed in Gurmukhi script: a hybrid approach. The Visual Computer, 37(7), pp.1637-1659.

[10] Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L. and Lopez, A., 2020. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408, pp.189-215.

[11] Ahmad, A.R., Khalia, M., Viard-Gaudin, C. and Poisson, E., 2004, November. Online handwriting recognition using support vector machine. In 2004 IEEE Region 10 Conference TENCON 2004. (pp. 311-314). IEEE.

[12] Balaha, H.M., Ali, H.A., Saraya, M. and Badawy, M., 2021. A new Arabic handwritten character recognition deep learning system (AHCR-DLS). Neural

Computing and Applications, 33(11), pp.6325-6367.

[13] Dey, R., Balabantaray, R.C. and Mohanty, S., 2022. Offline Odia handwritten character recognition with a focus on compound characters. Multimedia Tools and Applications, 81(8), pp.10469-10495.

[14] John, J., 2021. Support Vector Machine for Handwritten Character Recognition. arXiv preprint arXiv:2109.03081.

[15] Alkilani, A.H. and Nusir, M.I., 2021, November. Off-line Handwritten Verification Model for Processing Bank Checks Based on Truncated-SVD and Support Vector Machine (SVM). In 2021 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) (pp. 37-42). IEEE.

[16] Prashanth, D.S., Mehta, R., Ramana, K. and Bhaskar, V., 2022. Handwritten Devanagari Character Recognition using modified Lenet and Alexnet convolution neural networks. Wireless Personal Communications, 122(1), pp.349-378.

TABLE II. THE RESULT OF THREE DATASETS USING SVM

Data Set	Samples	Epsilon set tolerance of termination criteria	Number Support Vector (SV)	Accuracy
TD1 (Lowercase)	189,411	0.07	58,563	86.0077% (32578/37878)
TD2 (Uppercase)	217,812	0.02	53,627	88.4671% (43095/48713)
		0.13	53,662	88.4671% (43095/48713)
TD3 (Lowercase+ Uppercase)	407,223	0.004	179,943	73.4464% (63598/86591)