Multi Lingual Character Recognition using Hierarchical Rule Based Classification and Artificial Neural Network

Dr. Anupam Shukla[1], Dr. Ritu Tiwari[2], Anand Ranjan[3], Rahul Kala[4],

[1,2,3,4] Department of Information Technology, Indian Institute of Information Technology and Management, Gwalior, MP, INDIA
[1] dranupamshukla@gmail.com, [2] rt_twr@yahoo.co.in, [3] anand.iiitm@gmail.com, [4]rahulkalaiiitm@yahoo.co.in

Citation: A. Shukla, R. Tiwari, A. Ranjan, R. Kala (2009) Multi Lingual Character Recognition using Hierarchical Rule Based Classification and Artificial Neural Network, *Proceedings of the Sixth International Symposium on Neural Networks, Springer Verlag Lecture Notes in Computer Science*, Wukan, China, pp 821–830.

Final Version Available At: http://link.springer.com/chapter/10.1007%2F978-3-642-01510-6_92

Abstract. Optical Character Recognition is one of the rapidly growing areas of Artificial Intelligence due to its vast applicability. The technique is used to recognize characters printed on paper or elsewhere. The optical character recognition gains more importance when there are multiple languages present. The complexity of the problem increases for the addition of every language. The identification of character is both difficult and important in the presence of multiple languages. In this paper we propose a technique for multi lingual character identification. The algorithm uses the characteristics of the language to find out the language first. This is done using a rule-based approach. Then we apply neural network of the particular language to find out the exact character. We have coded and tested this using English capital letters, English small letters and Hindi letters. We got an appreciable accuracy using test cases of all languages. This proves the efficiency of the algorithm.

Keywords: Optical Character Recognition, Artificial Neural Network, Rule Based Approach, Classification.

1 Introduction

Optical Character Recognition refers to the identification of character that is written or drawn. The identification involves the classification of the given input into all the types of characters possible.

There are two types of recognition techniques possible, offline and online. In offline recognition technique, the final character is available that we have to identify [1]. We have no idea as to how this character was formed. On the other hand, in online character recognition, the character formation is known. The character is formed while the algorithm runs. Hence, we may sample the input in spans of time to get an idea as to how the character was formed.

The algorithm uses a hierarchical approach. First a rule based approach is applied to try to find out the language that the character belongs to. This is done by analyzing the inputs against the set of rules defined. The second step is to use the neural network of the identified language to identify the particular character.

Rule based approach works on the principle of firing a fixed set of rules on the input. The rules are formed and clearly stated at the time of design. The result of these rules decides the output.

Artificial neural network is a good means of machine learning. They are helpful for the system to learn from the historical data. Once it has learned, it is used for the testing purposes. The neural network, used in this paper is especially useful for the classification problems, where we create different classes of possible outputs and the net result is the cumulative result of these classes.

In this paper, we have made different neural networks for two languages (English and Hindi). These neural networks are trained from the training data of the respective languages. We take an input and apply a rule based approach first. The rule based approach finds out the language that the given character belongs to. This is then fed into the neural network of that particular language. This way we are able to recognize the character.

This paper is organized as follows. Section 2 talks about present works and motivation for the problem. Section 3 presents the algorithm and discusses its various aspects. Section 4 presents the results. Section 5 gives the conclusion.

2 Motivations

The problem of character recognition is one of the most rapidly developing fields. A lot of contribution in the field comes from Artificial Neural Network. A lot of work is also done using Hidden Markov Models (HMM). Various Genetic Optimizations have been applied to this problem with reference to feature extraction, matching etc [2, 3].

Neural Networks are a good means of learning from the historical data and then applying the algorithm to a new input [4, 5, 6, 7, 8, 9]. The basic approach in this method is to break the sample given into a grid. Each member of this grid is either 0 or 1 depending whether there was something written on that position. The neural networks are applied straight on the input that gives us the output.

Hidden Markov Model (HMM), on the other hand, is completely statistical model [10, 11]. This model is used to find out the probability of the occurrence of any character, based on the given input. This method is used extensively for these kinds of problems where a classification needs to be done.

Rule based approach has been applied to various problems. It is a good approach and works very well when it is possible to write down such rules. The rules may be written if they are simple to write and less in number. Usually these rules would be written when they are distinctly visible. Many classification problems have been solved sung this approach.

3 Algorithm

In this section we would discuss the details of the algorithm. The general algorithm involves segmentation, preprocessing and recognition [12, 13]. These steps are applied one after the other. Our algorithm assumes that segmentation and preprocessing have already been done. The final phase is the recognition phase, where the characters are recognized. We discuss the recognition algorithm in the following sub-sections.

3.1 Hierarchical Nature

In this algorithm we have used a hierarchical structure comprising of the neural networks and rule-based approach. A combination of these is used for the identification of the character.

We have used capital English letters, small English letters and Hindi letters for this algorithm. The general structure of the algorithm is given in Fig. 1. In this algorithm, the input is a matrix of 14X14. Each of these represents a pixel. The pixel is marked as 1 or -1 depending on whether something was written in the area of that pixel or not. If something was written, the pixel is marked as 1; else the pixel is marked as -1.

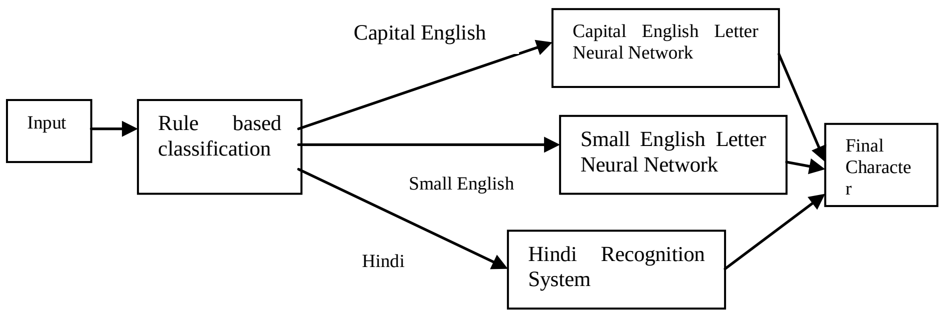

Capital English
Capital English Letter Neural Network
Input
Rule based classification
Small English Letter Neural Network
Final Character
Small English
Hindi
Hindi Recognition System

Fig. 1. The general recognition algorithm

The algorithm uses a set of rules to decide the language of the character given as input. If the given input is found to be English capital letter, the original input matrix is scaled down to a scale of 7X7. This may be done by deleting alternative rows and columns. If the row/column being deleted represents a vertical/horizontal line, the other row may be deleted. This is given as the input to the Neural Network of the English capital letters. If the given input is of English small letters, a similar procedure is followed. The whole process is of breaking the matrix into smaller size of 7X7and giving it to the neural network of the English characters. If the given input is of Hindi characters, we do not scale down the original image. Rather we break it into sections. These sections are made based on the philosophy of the Hindi characters. There is a top section that is used to accommodate the overhead projection of the vowels. Similarly there is a left, right and bottom section. These are used to accommodate their respective projections. These projections decide the character from their own way. The leftover space at the center is for the main character. This is shown in Figure 2(a), 2(b), 2(c), 2(d) and 2(e). Each of these neural networks takes the given input and results are shown. The result is the identified character.

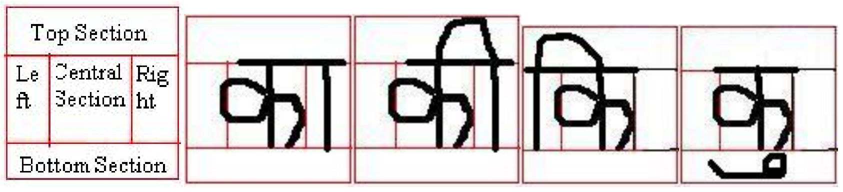


Fig. 2(a), 3(b), 3(c), 3(d) and 3(e). Segmentation of the Hindi characters

3.2 Rule Based Approach

In this approach we keep firing the rules, one after the other, to identify the final class to which the output belongs. This approach may be taken as a "if…. else if….. else" statements. The general structure of this kind of approach is

```
If condition$_1$ $\rightarrow$ result$_1$
Else if condition$_2$ $\rightarrow$ result$_2$
Else if condition$_3$ $\rightarrow$ result$_3$
….
….
Else result$_n$
```

Classification. In order to write the rules, we first classify the various inputs as classes. We have the major three classes of inputs as mentioned below:

Class A: Capital English letters
Class B: Small English Letters
Class C: Hindi Letters

These are further classified according to the characteristics within the language. While writing the rules, we would be making use of these classifications. The classes of the Capital English letters are:

Class A-1: A, B, D, E, F, I, J, K, M, N, P, R, T, Y
Class A-2: C, O, S, U, V, W, X, Z
Class A-3: E, F, I, T, Z
Class A-4: H, L
Class A-5: G, Q
Class A-6: J

The classes for the small English characters are as follows:

Class B-1: b, d, h, i, j, k, l, p, t
Class B-2: a, e, f, g, m, n, q, r, x, y
Class B-3: c, o, s, u, v, w, x, z

For the Hindi characters, we have just one class. This class comprises of all the characters found in Hindi and also incorporates all the combinations allowed by the language.

Class C: All Hindi characters

Now we have the various classes and subclasses are ready. We observe the classes to find out the common characteristics of the classes. We may exploit these characteristics at the time of writing down the rules. The characteristics are presented as under:

- All the characters of class C have one thing in common. They have a line at the top. In Hindi language, this line is always present and may be regarded as the base line. This line may not be present at the start, but is always present at the end. This is shown in figure 3(a) and 3(b). Further it is not possible to make any character in Hindi with the use of lines alone. We would be required to use curves for the purpose of writing the character.

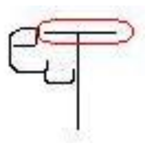


Fig. 3(a). Top line half

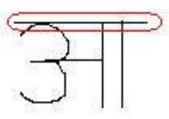


Fig. 3(b). Top line full

- Class A-3 and A-6 are the only characters of the Class A that have a horizontal line at the top. In rest of the characters, there is no such horizontal line at the top.
- More specifically, A-3 is the class that has a horizontal line at the top and the whole class is a set of lines only with no curve. On the other hand class A-6 has a horizontal line with a curve at the bottom.
- Class A-2 and B-3 are the similar classes. Here the character is written in the same way in English capital and English small. As a result we may not identify whether the character belongs to the A-2 class or B-3 class. Both the answers would be permissible to the further algorithms.
- Class B-1 is the set of characters in small English characters, where there is a small vertical line at the top. This is a very small line of more than 1/3rd the length of the total character. This line is shown in figure 4.

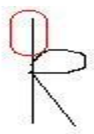


Fig.4. the top line in B-1

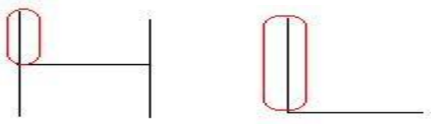


Fig 5(a), 5(b): the vertical line at Class A-4

- Class A-4 is the only set of characters in English capital characters that have a vertical line at the top. This line is shown in figure 5(a) and 5(b).
- Class A-1 is the set of characters in capital English characters where there is always a straight line that joins the topmost and leftmost point to any other point. This line is shown in figure 6(a) and 6(b).

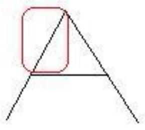


Figure 6(a): The straight line in A-1

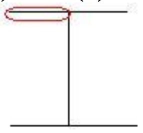


Figure 6(b): The straight line in A-1

- Class A-5 is the class which does not contain a line from the topmost and leftmost point. Rather it is predominantly a curve.

Tests. Having this background and properties of the various classes and the properties, we write down the various tests. These tests would be used to exploit the characteristics of the individual classes. Hence using these tests, we may be able to find out the class to which the input belongs.

In order to facilitate the various tests, we develop a concept of traversal of the input grid. We try to travel the grid using only the points that are marked as ‘1’. Hence we use the points where there was something written in the input character to be recognized. We do not traverse on any point more than once. This means that once we have stepped on any coordinate, we cannot move on it again. The traversal always starts from a specific coordinate. This coordinate has to be specified at the start. Whenever there is only one point where a move can be made, we assume that we are moving on a line or a curve present in the character. On the other hand, if there is more than one possibility to make a move, we assume that we have an intersection and the point where the algorithm is the intersection point. This traversal, however, needs to be performed on a high resolution image or grid. Also there must be no noise for the algorithm to work correctly.

The general algorithm of this traversal is given in Fig. 7. We treat the algorithm like a graph algorithm. Here the coordinates are vertices and the possibility to move from one vertex to the adjacent vertex act as edges. We take all the coordinates where the algorithm traversed. This is used for its analysis for possibility of lines and curves.

We use another algorithm for line detection. This algorithm takes its input as a set of coordinates, we had traversed on. These set of coordinates are then analyzed. For these set of coordinates to be a line, the angle between the first point, the last point and the middle point must be almost 180 degrees. We use the cosine rule of triangles for finding the angle. The angle is given in Fig. 8. For this line detection, it is desired that the set of coordinates have sufficiently high number of points.

Class C Test: In this test we check for the top vertical line in the character. The general way is to do this test is to find out the rightmost point in the top 1/3rd of the input grid. This can be taken as the rightmost point in the horizontal line that we wish to find. Starting from this point, we keep traversing till a point of intersection is met. Using these set of coordinates we decide whether it is a line. If these set of coordinates come out to be a line, we say that the test was positive. If on the other hand, the set of points do not lie on a line, we say that the test failed.

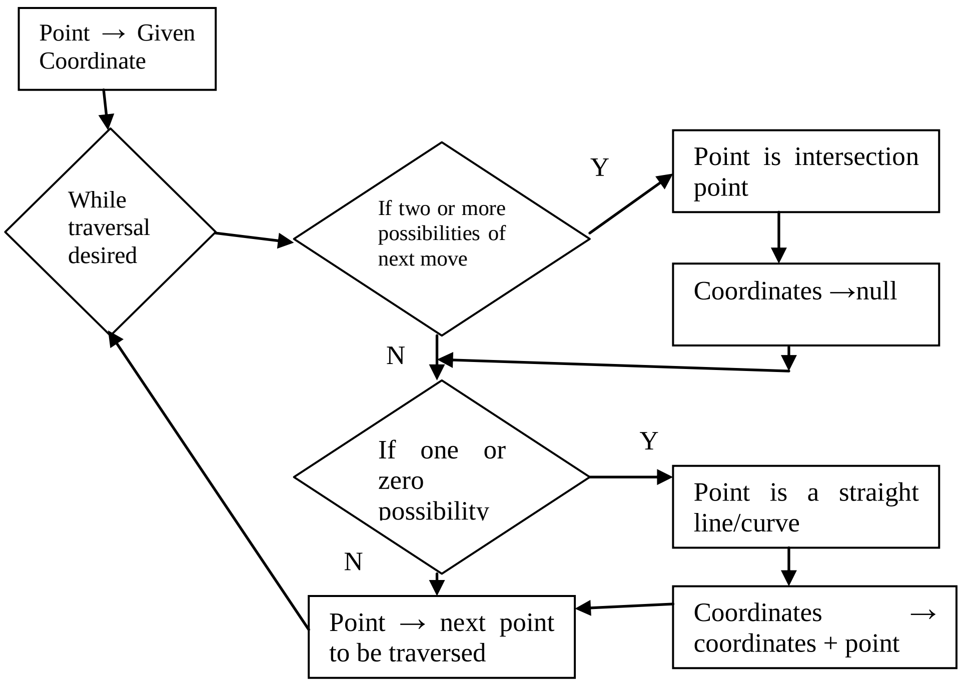
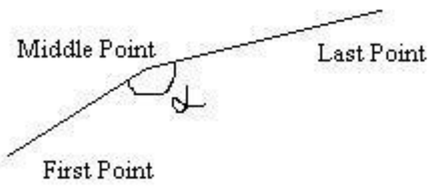

Point → Given Coordinate
While traversal desired
If two or more possibilities of next move
Y
Point is intersection point
Coordinates →null
N
If one or zero possibility
Y
Point is a straight line/curve
N
Point → next point to be traversed
Coordinates → coordinates + point

Fig. 7. The traversal algorithm



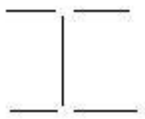


Fig. 8. The angle for line detection **Fig. 9.** The 5 different paths for I

Class A-3 Test: In this test we check whether the figure contains only lines, or it contains both lines and curves. We start from the topmost and leftmost point. Using this point we start traversing. At every intersection, we traverse through all the points possible. Each set of paths traversed are recorded as the set of coordinates. At the end of this procedure, we clearly know all the set of paths that may be lines or curves. We apply the algorithm to each and every path to find out whether it is a line. If all the paths are found to be lines, we say that the Class A-3 test has been passed; else we say that the class A-3 test is failed. The 5 paths generated when ‘I’ was the input is given in figure 9.

Class A-6 Test: The purpose for this test is to test whether the given input is J. This test checks for the presence of one top line and one curve at the bottom. Both these may be easily done using information provided in the Class C test and Class A-3 Test.

Class B-1 Test: This test checks for the presence of a small vertical line of at least 1/3rd length of the whole character at the top of the character. We find out the topmost and the leftmost point in the given input. We start traversing from this point. We need to make sure that we travel in almost bottom direction. Once the intersection point is reached, we break. We also check the total length of the path travel and whether it was a line or not. If both the conditions satisfy, we say that Class B-1 test succeeded.

Class A-4 Test: The purpose of this test is to check for the availability of H and L. These are the two characters with a vertical line and without curves. The test is similar to the B-1 test. Here we also apply a test similar to A-3 test to find 5 straight lines for H and two straight lines, one vertical and one horizontal (or 1 curve) for L. When these conditions are met, we say that the Class A-4 test has passed.

Class A-1 Test: In this test, we check for the presence of a straight line traversing from the top left point. This test is similar in nature to the Class C test.

*Class A-5 Test***:** In this test we check for the presence of G and Q. These are predominantly curves in nature. If we traverse leftward from the top left position. We will find a curve in both of these figures. This curve would be intercepted by a straight line. This may be used to distinguish the class.

Rules. Once we have specified all the tests, we need to write a rule that would cover all the different classes and would classify the different inputs to their correct classes. Our aim is to map the input to class A or class B or class C.

The rule for this problem may be stated as:

If Class C test is positive and Class A-3 test fails → Class C
Else if Class C test is positive and Class A-3 test is positive → Class A
Else if Class C test is positive and Class A-6 test is positive → Class A
Else if B-1 test is positive and A-4 test is negative→ Class B
Else if B-1 test is positive and A-4 test is positive → Class A
Else if A-1 test is positive → Class A
Else if A-5 test is positive → Class A
Else Class B

Using this rule, we may be able to find out the class for any input. If we take any character and apply to this rule, we would find that this rule correctly maps the input to the correct class. Hence we have made a system where we come to know the language for any given character.

3.3 Artificial Neural Network

Once we have known the class to which the input belongs, we need to apply the input to the artificial neural network of the particular language. The artificial neural network of the English capital and English small letters are identical. On the other hand the artificial neural network of the Hindi characters is different. Here different neural networks are applied to the different segments of the image. We have a neural network for the top segment, one for the bottom and one for the center. There are only two possibilities in the right and left segment. Either there would be a vertical line or nothing. Hence we do not make a neural network for these two segments. The specifications of the neural network are given in the subsequent sections.

Outputs. In this problem we have used a special kind of neural network; specially designed for the purposes of the classification problems. Both capital and small English has 26 such output classes. The number of outputs in our neural network is the number of such output classes. For any input i, there are same number of outputs as there are number of classifying classes. Each of these outputs shows the possibility of the input belonging to that class. This value lies between -1 and 1. 1 is the highest probable and -1 the least. Hence after the application of any input i, we find out the output with the maximum possibility. This is regarded as the final output.

Let’s say we have the neural network of the capital English characters. It takes a grid of 7X7 as input. Hence there are 49 inputs <I_1, I_2, I_3, I_4….. I_{49}>. Corresponding to this there can be 26 classes possible as outputs (A-Z). The output array would be in form of <O_1, O_2, O_3, O_4….. O_{26}>. Here O_1 corresponds to A, O_2 to B, O_3 to C and so on. If we are giving the training data for B, we know that the possibility of O_2 will be 1 and the possibility of any other output will be -1. Hence the training data output for B will be <-1, 1, -1, -1 ….. -1>. While testing we find the highest output, max(O_i) for all i from 1 to 26. The character corresponding to this is the final answer.

Identification. In capital English characters and small English characters, we straight away get the final answer. But in Hindi character recognition, we get the presence of various signs at the top, bottom, right, lefty and center positions. Once it is known what lays at all the positions, we may easily come to know the final character (character and the vowel associated).

4 Results

In order to test the system, we coded the system in MATLAB. The input was made and fed into the system as training data. Separate databases were made for the various Artificial Neural Networks.

The first neural network was trained for capital English letters. It had one hidden layer with 300 neurons, 26 neurons in output layer and 49 neurons in input layer. The second neural network for the English small characters had a similar configuration except for that there were 100 neurons in the hidden layer. The main Neural Network for Hindi characters had 49 input neurons, 100 neurons in the hidden layer and 33 output neurons. The results of the algorithm are given in table 1.

Table 1. Results of the algorithm

Total Test Cases	*218*
Total Correct:	*193*
Total Incorrect:	*25*
Efficiency:	*88.53*

An efficiency of 88.53% clearly shows that the method works with high efficiency. Hence we have been able to build a multi lingual character recognition algorithm.

5 Conclusions

In this paper we proposed an algorithm that solved the problem of multi-lingual character recognition. The algorithm uses a hierarchical approach for the problem. The first step is to use a rule-based system. This system takes an input and applies standard rules to the input to get the output class. Once the class is known, we apply the given input to the class specific neural network.

While we have presented good algorithms for the detection of lines and curves, better and more robust algorithms and logic may be developed in the future. More languages may be added in this system and consequent changes in rules may be made.

References

1. Draghici, S.: A neural network based artificial vision system for licence plate recognition. In: International Journal of Network Security, International Journal of Neural Systems, Vol. 8, No. 1, 1997
2. Liwicki, M., Bunke, H.: Handwriting Recognition of Whiteboard Notes: In: ieeexplore 29 Aug.-1 Sept. 2005
3. Soryani, M., Rafat, N.: Application of Genetic Algorithms to Feature Subset Selection in a Farsi OCR. In: Proceedings of World Academy of Science, Engineering and Technology Volume 18 December 2006 ISSN 1307- 6884
4. Graves, A., Fernandez, S., Liwicki, M., Bunke, H., Schmidhuber, J.: Unconstrained Online Handwriting Recognition with Recurrent Neural Networks. In: Advances in Neural Information Processing Systems 2008
5. Bertolami, R., Zimmermann, M., Bunke, H.: Rejection strategies for offline handwritten text line recognition. In: ACM Portal, Vol. 27, Issue. 16, (December 2006)
6. Shi, D., Shu, W., Liu, H.: Feature Selection for Handwritten Chinese Character Recognition Based on Genetic Algorithms
7. Yuelong, L., Jinping, L., Li, M.: Character Recognition Based on Hierarchical RBF Neural Networks. In: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06) -Volume 01
8. Araokar, S.: Visual Character Recognition using Artificial Neural Networks
9. Gunter, S., Bunke, H.: Off-line Cursive Handwriting Recognition - On the Influence of Training Set and Vocabulary Size in Multiple Classifier Systems. In: Elsevier Volume 43, Issues 3-5, March-May 2005, Pages 437-454

10. Hewavitharana, S., Fernando, H. C., Kodikara, N.D.: Off-line Sinhala Handwriting Recognition using Hidden Markov Models.
11. Rao, P., Shankar, Aditya, J.: Handwriting Recognition – Offline Approach.
12. Som, T., Saha, S.: Handwritten character recognition by using Neural-network and Euclidean distance metric. In: Social Science Research Network
13. Gernot, A., Fink, P., Thomas: Unsupervised Estimation of Writing Style Models for Improved Unconstrained Off-line Handwriting Recognition.
14. Chellapilla, K., Larson, K., Simard, P., Czerwinski, M.: Computers beat Humans at Single Character Recognition in Reading based Human Interaction Proofs (HIPs). In: Proceedings of the Second Conference on Email and Anti-Spam (July 21-22, 2008)