Word Spotting and Character Recognition using Quadrant Density and Aspect Ratio

Nivas K S
Dept. of Information Technology,
SNS College of Engg.
Coimbatore.

Preethi V
Dept. of Information Technology,
SNS College of Engg.
Coimbatore.

ABSTRACT

The desire to access, search and explore large amount of documents had paved the way for digitizing and storing the document in computer for easy access. But doing a word search (word spotting) in a scanned document is difficult, since the entire document is saved as an image. Different methods are already proposed which recognizes the characters from the scanned document and converts it into a text document in which the word spotting is done. In proposed method, a set of features are extracted from each character and the feature vector is converted to a floating point value. This floating point is a combination of quadrant densities obtained from the character and its aspect ratio with their respective importance. Now using this floating point recognizing the character with reference to a trained set of floating point values can be done. Now when the user searches for a certain word, spatial adjacency algorithm is used to spot the searched keyword directly in the image.Character recognition is one of the most dynamic part of today's artificial intelligence systems. Here the proposing system analysis the similarity between various characters in a language and trains itself to identify and understand similar characters using the previously learned data.

1. INTRODUCTION

1.1 Need for the System Word spotting

For example to digitize the book of Joanne Rowling's Harry Potter a full night typing effort is required. Else by using use a high end Scanner and in minutes scan all of seven books into the computer using the character recognition technology.

If the scanned images are manuscripts, The amount of variability in handwriting and the high noise levels in manuscript documents are currently transcribed by hand. So to rectify this type of problem word spotting using arithmetic coding needs to be considered.

1.2 Need for the system Character recognition

This technology is about Handwriting recognition and the technique is about the Digitization and character recognition of manuscript. The technique will give importance to the handwriting recognition research; it gives relatively new view on the data and thus can be an addition to the existing systems. It can be combined with the other systems. The way the system works adds number of possibilities for applications of character recognition, and the proposed system is much faster than the existing systems.

1.3 Objectives

The goal of the word spotting idea applied to the scanned documents is to greatly reduce the amount of annotation work that has to be performed by grouping all words into clusters. Ideally each cluster contains words with the same annotation.

The vital role of the proposed method will give importance to the character recognition research; it gives relatively new view on the data and thus can be an addition to the existing systems. Hope that the given technique will be much faster than comparing to the other techniques as the final phase of the recognition is just an arithmetic calculation.

2. LITERATURE SURVEY/BACKGROUND

2.1 Existing Character Recognition system

Optical character recognition (OCR) technology is the first method to recognize the character in the given input hand written image. The source material must be scanned using an optical scanner to read in the pages as bitmap. OCR will differentiate between images, text and determine what letters are represented in the light and dark areas.

The OCR can identify a wide variety of fonts, but handwriting and script fonts that mimic handwriting are still problematic.

2.2 Existing Word Spotting System

The goal of the word spotting idea applied to the hand written documents is to greatly reduce the amount of annotation work that has to be performed by grouping all words into clusters. Ideally each cluster contains words with the same annotation once such a clustering of the data set exits, the number of words contained in a cluster can be used as a cue for determining the importance of the word as a query term. For example, highly frequent terms, such as stop words can be discarded. All clusters with terms that are deemed important can then be manually annotated. This makes it possible to construct a partial index for the analyzed document collection, which can be used for retrieval.

2.3 Motivation for New Method

Different methods are already proposed which recognizes the characters from the scanned document and converts it into a text document in which the word spotting is done. In this project a more efficient algorithm is proposed which does the word spotting in the feature extraction phase itself.

Hence a system is proposed that analyses the similarity between various characters in a language and trains itself to identify and understand similar characters using the

previously learned data. Here we have also taken up the effort of making the system faster and versatile for real time usage.

3. SYSTEM STUDY

3.1 Proposed Character recognition system

In this methodology quadrant density and aspect ratio of the character is used to recognize the character. This method needs a training set of characters to begin with.

1. Find the appropriate Training set.

2. Extract the distinguished feature of the character image and the training image, like Quadrant Density and Aspect ratio.

3. Finding the grand float or weighted float for the characters in the image which will distinctly represent each character.

4. Finding the closest or the most similar character with the help of the given training set (character recognition).

3.2 Proposed Word spotting system

Once the character recognition phase is over all the recognized character and their dimensions are made into a vector of user defined data structure. From this vector word spotting is performed using a spatial adjacency algorithm.

1. Obtain keyword from user.
2. Parse through the data structure vector to find relevant nodes.
3. Using spatial adjacency algorithm spot the word in the image.
4. Highlight found result.

3.3 Hardware specification

*512MB RAM

*10 GB Hard disk

*Scanner

3.4 Software Specification

***** Visual studio

* Open CV

* Windows OS

4. SYSTEM ANALYSIS AND DESIGN

4.1 Constrains

Analyzing the availability of a practical recognition system for handwritten characters without any constraints has a long way to go we have attempted to find a set of writing constraints which significantly improves the machine-readability. Based on the observation that the majority of the misrecognitions reported are caused by ambiguous characters, a set of writing constraints is developed which maximally disambiguate those characters. So based on experiments, we have confirmed that the recognition rate of those handwritten data could be improved significantly by applying the proposed set of constraints

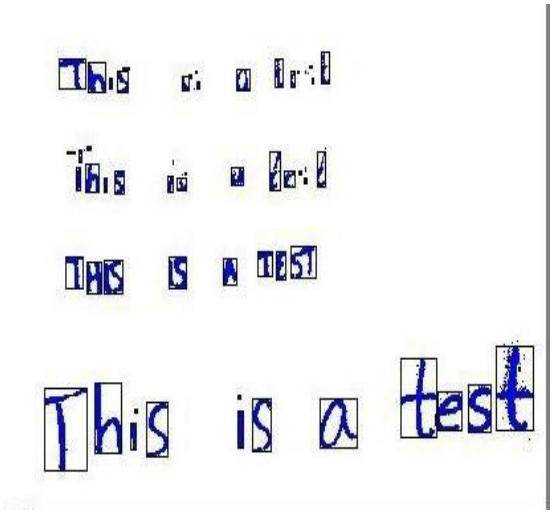
1. Noise level should be minimal.
2. The training image with images of all different alphabets is required.
3. Minimal deviation of font size is required.

5. IMPLEMENTATION DETAILS

5.1 Character Segmentation

The concept of Character segmentation is to decompose an image of a sequence of character into sub images of individual symbols. This sequence will continue until no additional character images are found.

The characters are segmented when any black pixel is found, the measure function will be called and then each and every character will be redrawn with blue instead of black.



5.2 Implementation of Character Recognition

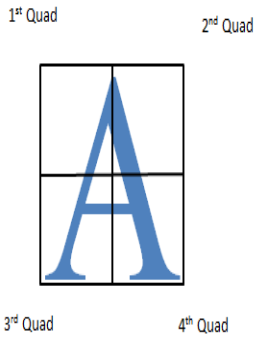
5.2.1 Training and Inputs

Any input image that contains data to be recognized as characters is taken. During the start of the process a training image is first required which contains all the elements in the character set so as to train the system for the process of character recognition. Both the training image and the input image will be subjected to processes like finding quadrant densities and aspect ratios in a similar way.

5.2.2 Finding Quadrants Density

Quadrant density is a variable quantity for each character in the English language. But same characters exhibit almost the same quadrant densities. So the concept to differentiate between two different characters is taken into account.

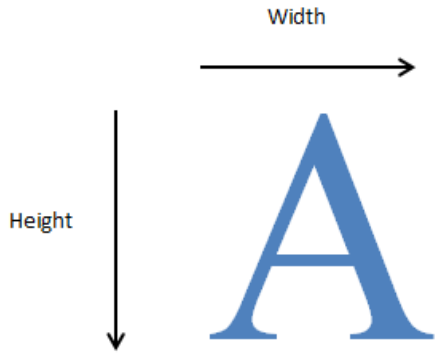
First split the whole character into four different quadrants by partitioning them at length/2 and width/2. Quadrant density is actually a measure of the amount of the character contained in that specific quadrant. Or in other words the percentage of that character that can be represented by that quadrant alone. That is found by the ratio between the numbers of pixels that is occupied by that character in that quadrant to the total number of pixels occupied by that character.



$$q = \frac{\text{Number of filled pixels in the qudrant}}{\text{Total number of filled pixels}}$$

5.2.3 *Finding the Aspect Ratio*

The Aspect ratio is a deciding factor of characters with similar quadrant density as it represents the orientation of the character. The aspect ratio is the ratio between the length and the width of the character.



$$\text{Ratio} = \frac{Height}{Width}$$

5.2.4 *Finding the importance of each measure*

So having all the measures of a character, before convert them into a single variable value, the importance or the vitality of each of these measures is decided.

Then find the collective quadrant densities of all the characters in the English language in order to analyze the character set of the language.

F1 (char) = q1
F2 (char) = q2

F3 (char) = q3
F4 (char) = q4

$$Q1 = \sum_{char=A}^{Z} F1(char) + \sum_{char=a}^{z} F1(char)$$

$$Q2 = \sum_{char=A}^{Z} F2(char) + \sum_{char=a}^{z} F2(char)$$

$$Q3 = \sum_{char=A}^{Z} F3(char) + \sum_{char=a}^{z} F3(char)$$

$$Q4 = \sum_{char=A}^{Z} F4(char) + \sum_{char=a}^{z} F4(char)$$

Q1, Q2, Q3, Q4 are the collective quadrant density of all the character in the English language.

$$Q = Q1 + Q2 + Q3 + Q4$$

$$Imp\,q1 = \frac{Q1}{Q} \qquad Imp\,q3 = \frac{Q3}{Q}$$

$$Imp\,q2 = \frac{Q2}{Q} \qquad Imp\,q4 = \frac{Q4}{Q}$$

Therefore the importance of quadrants have been found as follows,

Quad-1 22.2%

Quad-2 23.3%

Quad-3 26.6%

Quad-4 27.9%

5.2.5 Find the grand float or weighted float

As knowing the importance of each measure, now find the weighted float by taking the product of the quadrant density of the current picture and the importance ratio. Assign a complete 100% to the aspect ratio as it acts as a deciding factor while recognizing a character.

The weighted float is calculated by the formula is given below.

$$grandf = ratio*1 + q1*0.222793 + q2*0.232502 + q3*0.266360 + q4*0.271425 ;$$

5.2.6 Find the Close set

The results that obtained from the above step from the training image will act as trained set and the results that obtained from the input image will act as the input data that will be compared with the training set.

Here find the Euclidian distance between a picture element and each of the items in our training set. Then find the training cell with the least distance of all the others and thus find the most similar character, using the index of our training set with the least difference output the recognized character.

5.3 Implementation of Word Spotting

5.3.1 User Input

The first step of the word spotting is to get the required keyword from the user. Only after getting the keyword from the user we will be able to proceed to the word spotting phase.

5.3.2 Spotting the word

Word spotting is done by a recursive algorithm. The parameters passed to the function are the keyword and the vector of nodes, where each node represents a single character in the image.

The algorithm starts from the first character of the keyword and proceeds till the last character of the keyword and in each step it finds nodes from the vector of nodes that contains the same character in the keyword. Once it finds that the character in a certain node is same as the next character in the keyword, it passes both of these nodes to a function which checks if they are adjacent to each other. To find if the found node is actually a right hit, the adjacency function performs three different evaluations on the given nodes.

Right Adjacency

This function checks if the newly found character lies to the right of the current node. Only if the new node lies to the right of the current node it is a valid hit otherwise it's an invalid hit.

This phase also checks if there is any other character that lies in between these nodes and those conforming the fact that they are indeed adjacent.

In Same Line

The next criterion is that the newly found character should lie in the same line. To check if the character lies in the same line, find the centroid of the current character and use that to check if the newly found character is in the same line.

Not the same

The next criterion is to check if the newly found node is not as same as the current one. This acts as a failsafe. This action cannot be done by a simple conditional statement so it is implemented as a function to check if all values in the object are identical to each other.

5.3.3 Storing the found nodes

Whenever reach a point where get a correct hit, store that specific node so as to produce it as the result of the word spotting process. This is done by storing all the correct nodes in a vector of nodes. Each time reaching the last character of the keyword and perform a successful hit, pass this vector of nodes to a function that produces the result of the process.

5.3.4 Displaying the result

Once have a vector of nodes which form the keyword in the image, parse through these nodes to find out the maxima and minima of these nodes with respect to the image, i.e., find four extremes of the nodes: Minimum X, Y and Maximum X, Y.

Once find these extremes exhibit the result either by highlighting the background of the sub image or drawing a bounding box around the found sub image.

5.4 Accuracy for Word spotting

The word spotting algorithm is based on the character recognition algorithm. Hence the accuracy is same as the character recognition. If the accuracy of character recognition is 100% then the accuracy of the word spotting algorithm will also be 100%.

6. CONCLUSION AND FUTURE ENHANCEMENT

Our journey with specifying systems with Character Recognition and Word Spotting have enabled us to understand the science behind Digital image Processing and its applications. We took the case of a Scanned Image documents because the other software specification was varied for faults and inconsistencies by recognizing and checking certain temporal properties using the Digital image Process classical Methods

7. REFERENCES

[1] Stefan Klink, German Research centre for the Artificial Intelligence (DFKI,GmbH),Germany

[2] Andreas Dengel, Thomas Kieninger German Research canter for the Artificial Intelligence (DFKI, GmbH), Germany

[3] Sami Lais is a freelance writer in Takoma Park, Md.

[4] T.M.Rath, R.Manmatha et al.[trath,manmatha] @ cs.umass.edu

[5] T.M.Rath and R. Manmatha, "Word spotting for historical documents", *International Journal on Document Analysis and Recognition* (*IJDAR*), Vol.9, No 2 – 4, pp. 139– 152 , 2006.

[6] V. Lavrenko, T. M. Rath, R. Manmatha: "Holistic Word Recognition for Handwritten Historical Documents", *Proceedings of the First International Workshop on Document Image Analysis for Libraries (DIAL'04)*,pp 278- 287, 2004.

[7] Niblack, W., "An Introduction to Digital Image Processing", pp.115–116. Prentice Hall, Englewood Cliffs, NJ, (1986).

[8] A. Bhardwaj, D. Jose, and V. Govindaraju. Script independent word spotting in multilingual documents. 2nd Intl Workshop on Cross Lingual Information Access, pages 48–54, 2008

[9] A. Fischer, A. Keller, V. Frinken, and H. Bunke. Lexicon-free handwritten word spotting using character hmms. Pattern Recogn. Lett., 33(7):934–942, May 2012.

[10] R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal. Database development and recognition of handwritten devanagari legal amount words. Document Analysis and Recognition, International Conference on, 0:304–308, 2011.

[11] R. Saabni and J. El-Sana. Keyword searching for Arabic handwritten documents. 11th International Conference on Frontiers in Handwriting recognition (ICFHR2008), pages 716–722, 2008.

[12] S. N. Srihari, H. Srinivasan, C. Huang, and S. Shetty. Spotting words in latin, devanagari and arabic scripts. Artificial Intelligence, page 2006.