Proposal of character recognition algorithm using database

Takahiro Fujishima, Ken Kariya, Lifeng Zhang

Kyusyu Institute of Technology, Kitakyushu, 800-0207, Japan

*Corresponding Author: zhang@ele.kyutech.ac.jp

Abstract

In recent years the paper-based information converted to digital text data and managed by the personal computer becomes popular. By converting it into data, companies can enjoy many merits such as cost reduction, improvement of business speed, avoidance of risks. One of the indispensable means for data conversion is OCR. OCR is an optical character recognition system widely used in general, but there are cases where false recognition is made depending on environment and equipment. OCR makes it easy to recognize the characters printed on a paper with a specific resolution. However, it is not easy to recognize letters for deformed characters and handwritten characters. Once the misrecognition happened, it might cause huge damage to businesses in some cases. In fact, printed characters may be deformed in FAX documents sent from overseas. Therefore, in this research, the goal is creating a character recognition system to read characters of deformed Fax documents correctly. We aim for 100% recognition rate using a database and improved OCR. The proposed system searches the OCR-recognized data in the database and acquires accurate delivery data. In this research, we proposed a pattern matching approach using the database.

1. Introduction

The transition from paper media to digital data is one of the tasks that many companies are working on. By digitizing documents (paperless), companies can enjoy many merits (such as cost reduction, improvement of business speed, risk management). Document digitization has been tackled by companies more than before, but it has greatly accelerated by the mobile use of enterprises due to the spread of smartphones and tablets in recent years.

There are various kinds of documents in the world, and it is

challenging to recognize characters with an accuracy of 100%. For documents printed or printed at regular pitches and documents with a good print quality, it is possible to recognize characters at an accuracy close to 100%, but it is possible to recognize documents printed or printed at uncertain pitches or documents with low-quality images (FAX, copied originals) Or the like, the character recognition rate may decrease.

2. Principle

2.1 Database

A database can be defined as a collection of things that store data having mutual relationships together to be usable for a plurality of applications, to eliminate harmful and unnecessary redundancy. The data is stored to be independent of the program using the data. A common and controlled approach is used to add new data, repair and retrieve data present in the database. Structurally remember the data to be the basis for developing future applications. If the database is structurally distinct, one system will contain a collection of these databases.

Also, a database that handles data based on a concept called a relational model, which is one of the database formats, is called a relational database. In a relational database, columns (columns) and records (rows) are given to data, and they are arranged and arranged in a table (table). Data can be easily extracted by rearranging the data around each column and record. Relational databases are currently the most widespread and often refer to relational databases in the case of databases, for example. The system for operating and managing the relational database is called a relational database management system (RDBMS). Typical relational database management systems include Oracle Corporation Oracle, Microsoft Corporation SQL Server, MySQL, SQLite, PostgreSQL distributed as OSS, and others.

Among them, SQLite is a database that just gathers files and has a very simple structure.

Because databases are running differently from COBOL, C language, Java, Python and other programming languages, extensions for connection to use the database in a programming language also can be used. Connect to the database using this extension module and communicate with the database using SQL. SQL is a database language for manipulating data created for relational databases, in response to instructions from users and systems, queries the relational database and returns the results. It does not compile but behaves like an interpreted language.

2.2 OCR

(i) Image capture

Input the document into the computer using a scanner.

(ii) Layout analysis

Find the character part from the text region, image region, and ruled line of the document and decide the order to read.

(iii) Cut out lines

Decompose line by line in the character area.

(iv) Character segmentation

Pay attention to one line cut out by cutting lines, and break up one line for each character. In this process, while moving the left and right lines from top to bottom, the number of intersections of this line with characters is counted, and a place where this value is 0 is judged as a character and character break.

(v) Character recognition

Regarding individual characters, in order to accurately perform character recognition on character size, letter type (Mincho type, Gothic type, textbook body, etc.), character collapse and blur fluctuation, normalization, features processing proceeds in the order of extraction, matching, knowledge processing.

2.3 Tesseract-OCR

Fig. 1 Flowchart

Tesseract assumes that the input is a binary image and a polygonal text region is defined. First, we perform a connected component analysis where the outline of the component is stored. This has the advantage that it is possible to detect the reverse text by simply checking the nest of the outline, and the number of children, and the outline of the grandchild, and to recognize it as in black and white text readily. At this stage, the outline is purely nested and gathered in the blob. Blobs are organized into text lines, lines and borders are analyzed for fixed pitch or proportional text. Text lines are divided into different words depending on the type of character spacing. Fixed pitch text is immediately broken by character cells. The proportional text is divided into words using explicit space and fuzzy space. Recognition proceeds as a two-pass process. In the first pass, we try to

recognize each word sequentially. Each satisfactory word is passed to the adaptive classifier as training data. By using the adaptive classifier, it is possible to recognize the text below the page more accurately. In the adaptive classifier, the second pass is executed on the page where the word which is not sufficiently recognized is recognized again. In the final stage, we solve fuzzy space and examine other hypotheses of height x to find a lowercase text.

2.4 Pattern matching

Pattern matching is a method of specifying whether or not a specific pattern will appear and where it appears when searching for data. For pattern matching of character strings, there are various character string searching algorithms such as KMP method and BM method in the search of fixed patterns. Many methods using regular expressions have also been proposed. When comparing the data in the database with the character string in the conditional expression, it is possible to perform pattern matching using two special characters '%' and '_'.

3. Experiment

Experiment is conducted according to the flowchart in the previous chapter. The input image uses the image shown in Fig. 2.

For character recognition (OCR), learning data was created using fax print characters using jTess Box Editor, and OCR software (Tesseract-OCR) suitable for fax characters was used.

- (i) The number of counts was set to 100 times. We verified by changing the number of characters extracted from the character string to 3 to 5 letters, and examined the search accuracy in each case. The number of executions of the program was 100 times.
- (ii) The number of counts was set to 1000 times. We verified by changing the number of characters extracted from the character string to 3 to 5 letters, and examined the search accuracy in each case. The number of executions of the program was 100 times.

Fig. 2 Input image

4. Result

To test the search rate of this proposal, a simulation has been performed with 100 and 1000 times votes by marching arbitrary selected characters respectively.

Table 1 Search rate for 100 times votes

Randomly selected Character numbers	3	4	5
Item names			
SIGNWEIGHT(HDG)	25	43	54
MA STAND-G	0	0	0
F-18AEM	100	100	0

Table 2 Search rate for 1000 times votes

Randomly selected Character numbers Item names	3	4	5
SIGNWEIGHT(HDG)	3	17	66
MA STAND-G	0	0	0
F-18AEM	100	100	0

Table 1 and Table 2 show the search rate of failed items. All the other product names appeared in Fig. 1 but not shown in these two tables obtained the search rate with a probability of 100%.

5. Consideration

In existing OCR, all product names were not correctly recognized, but in the recognition method with using the database developed this time, 15 items out of 17 items could be recognized correctly. However, the erroneous search count of "MA STAND-G" was 100% in all cases of experiments 1 and 2. This is probably due to failure to recognize "G" of "MA STAND-G." The experiment was carried out with setting the number of counts this time as 100 times, 1000 times. The processing time taken for search all entered product names is about 1 second at the time of 100 counts and about 4 seconds at the time of 1000 times count. There was a difference of about 3 seconds. In addition, the number of erroneous searches was also larger when counting 1000 times than when counting 100 times, compared with the case of extracting characters in each case. Also, although it was output correctly, there was a character string with a small number of counts ("F-18AEM"). This is also considered to be caused by the wrong recognition result by Tesseract-OCR.

6. Conclusions

In this research, we examined whether it is possible to capture the trade name data of the correct character string by using SQLite as the database and pattern matching the OCR product name with the product name data in the database. As a result, it was possible to search and output the correct item name data even with the product name containing the erroneously recognized character. However, it can't be said that the performance is good because it is not possible to perform a correct search for all product names. It is thought by increasing the number of arbitrarily selected letters, the search rate of the correct product name would rise, but the performance did not rise as the search time got longer. In the future, to improve the performance, it is thought necessary to change the filtering and binarization methods at OCR to improve the character recognition rate.

References

- (1) Yoshihisa Kunitomo and Misa Kubo: "Database [revised 2nd edition]", 1983
- (2) NJK corporation, Character recognition technology, <http://mediadrive.jp/technology/techocr07.html> (accessed 2018/1/15)
- (3) IT vocabulary dictionary, Optical character recognition, <http://www.sophia-it.com/content/OCR> (accessed 2018/1/15)