Handwriting Recognition

Pre-processing and Layout Analysis

Perttu Pitkänen

May 10, 2016

Abstract

Abstract text

Contents

List of Abbreviations and Symbols

OCR	Optical Character Recognition
HWR	Handwriting Recognition
HOG	Histogram of Ordered Gradients
k	Number of selected neighbouring elements in k-nearest neighbors algorithm

1 Introduction

Optical Character Recognition (OCR) is the process of analyzing a image input of text and recognizing and extracting the characters to digital from. More specific case of character recognition process is the task of handwriting recognition (HWR) which concentrates on analyzing human hand written characters instead of printed characters. The unpredictable nature of human handwriting can make the task more challenging ascompared to printed text. Most HWR systems can be divided into two recognition approaches: *online handwriting recognition* and *offline handwriting recognition*. Offline handwriting recognition means analyzing an existing static image for handwritten text. Online handwriting recognition, on the other hand, is about analyzing the handwritten text on input including strokes and their order for example in touch screen appliances such as smartphones and tablet PCs.

Handwriting recognition can be applied to many practical uses such as document digitizaton or novel human-computer interaction method. Handwriting has remained popular as a way to take notes and transfer information, even though increased popularity and technological advancements of handheld digital devices have made digital information saving increasingly convinient. The process of handwriting recognition is still undergoing development.

For reliable results the handwritten input image must be processed appropriately. This includes preprocessing and layout analysis of which aim to enhance the image quality for later processing and find the different bodies of text. There is no general consensus of which methods or algorithms give the best results. This research will experiment with different methods to gain insight on each method or algorithms strengths and weaknesses.

2 Background

For long writing has been an important way of communicating for humans. Advacements with personal computers has diversified the methods to store and display textual information which has brought up new challenges and problems considering the transformation between traditional information and digital data. One of these challenges is the process of digitizing written text to computer readable and editable form.

Textual information can be in diverse forms and styles. These styles include machine printed text and handwritten text. Different approaches must be used when digitizing aforementioned styles of text.

Plenty of research has been conducted and several systems have been implemented for the purpose of optical character recognition and handwriting recognition. These systems can have drastically different approaches for processing the data, even if the data is similar.

2.1 Writing Recognition

Typical OCR and HWR systems consist of three phases:

1. Preprocessing
2. Layout Analysis
3. Feature Extraction
4. Classification

In image preprocessing stage the quality of image is enhanced and the area of interest is located. Additionally layout analysis phase can be included into the preprocessing stage. The feature extraction stage captures the distinctive characteristics of the digitized characters for recognition. Lastly in during the classification stage the feature vectors are processed to identify the characters and words. Each of these stages reduce the amount of information to be processed at a later step. [4]

2.1.1 Preprocessing

At preprocessing stage the image is enhanced by applying varying filters, transforms and segmentation to it. For text recognition it is important to reduce noise from the image. This can be done with appropriate filter e.g. Wiener filter.

Most of OCR softwares require the image to be binarized before analysis. It is important for later stages of recognition process that the binarized image contains as little noise and irrelevant objects as possible. These irrelevant objects can be caused by for example uneven lighting, paper texture or other non-textual objects such as drawings. Binarization method should be chosen carefully as the input image's paper texture can vary a lot.

Additionally, other ways to enhance the image before analysis have been experimented. Pesch et.al. discussed how contrast normalization, slant correction and size normalization can be applied to improve the handwriting recognition results. [10]

2.1.2 Layout Analysis

Layout analysis is the process to find where the actual text is located and what kind of textual blocks the image contains. These textual elements can contain titles, columns, captions consisting of text lines and words. Handwritten text is more likely to contain full page width single column text compared to printed text where more complex layouts are found more often. However handwritten text is more unpredictable compared to printed characters as the handwritten words can often overlap or have varying lineskews between lines.

2.1.3 Feature Extraction

To differentiate between words or individual characters some features must be extracted. These features can then be later used with classification stage to classify the input text.

Several experiments have been conducted for appropriate features. For instance raw intensity values of selected component can be used. More sophisticated features include histogram of ordered gradients (HOG) [7]. S. Dalal et.al. have discussed other feature extraction methods such as horizontal and vertical projection histograms, parameters of polynominals i.e. curve fitting and topological features such as loops, end points, dots, and junctions. [6]

2.1.4 Classification

Lastly in the recognition process is the classification phase. The goal of classfication is to find the class in which the new input will be assigned and by doing that finding which is the most likely meaning of each particular component. In this case inputs are features extracted from word or character and classes are the corresponding words or characters respectively.

Often during classification machine learning approaches are used. Machine learning algorithms look for repeating patterns in feature space and makes decisions and predictions according to those patterns. Common for machine learning algorithms is that they require some preliminary data to be processed in order to make later classification more robust. Machine learning algorithms that can be applied to text recognition include:

- Artificial neural networks
- K-Nearest Neighbor
- Hidden Markov model
- Support vector machine
- Recurrent neural networks
- Deep feedforward neural networks
- Decision tree learning
- Random forests

Simple example for machine learning is the k-Nearest Neighbor algorithm. The algorithm searches for the closest match of test data in the feature space. The previous training data is distributed in the feature space and classified accordingly. Specified amount of the nearest neighbors of the new node are counted and compared. The class has the most representation within these points is the class of the new node. The k stands for the amount of neighboring data points that are compared to the test data, and it should be declared as an odd number to prevent a tie from happening between two classes. The feature space can be constructed from feature vectors aquired in the previous phase. In general all machine learning algorithms work better when there are more feature dimensions, but this will result in a slower execution time.[sources??]

2.2 State of the Art

The subject of optical character recognition and handwriting recognition are widely researched subjects and well-functioning as well as feature rich software already exist. Many of these softwares utilize machine learning and neural networks to get satisfactory results especially with handwriting recognition. Many of the best OCR softwares are proprietary, thus making them unable for free research and analysis. Such software are for example Evernote which has an inbuilt OCR engine for searching text from pictures [9] and Abbyy FineReader software made especially for OCR [1]. Examples of open-source OCR software are previously mentioned Tesseract[11], OCRopus[3], Ocrad [8] and CuneinForm[5]. These pieces OCR software are not capable of handwriting recognition by default. Additionally, handwriting recognition algorithms developed by Jrgen Schmidhuber's research group at the Swiss AI Lab IDSIA have won several international handwriting competitions[2].

3 Implementation

implmnt

4 Evaluation

evltn

References

[1] ABBYY. ABBYY FineReader.

[2] A. D. Angelica and J. Schmidhuber. How bio-inspired deep learning keeps winning competitions. `http://www.kurzweilai.net/how-bio-inspired-deep-learning-keeps-winning-competitions`.

[3] T. Breuel. Announcing the OCRopus Open Source OCR System. `https://web.archive.org/web/20150313051644/http://googlecode.blogspot.com/2007/04/announcing-ocropus-open-source-ocr.html`, 2007.

[4] M. Cheriet, N. Kharma, C. Liu, and C. Suen. *Character recognition systems: a guide for students and practitioners.* 2007.

[5] Cognitive Technologies. CuneiForm. `http://cognitiveforms.com/products{\_}and{\_}services/cuneiform`, 2016.

[6] M. S. Dalal. A Survey for Feature Extraction Methods in Handwritten Script Identification.

[7] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 886–893, 2005.

[8] Free Software Foundation. Ocrad - The GNU OCR. `https://www.gnu.org/software/ocrad/`, 2016.

[9] B. Kelly. How Evernote's Image Recognition Works. `https://blog.evernote.com/tech/2013/07/18/how-evernotes-image-recognition-works/`.

[10] H. Pesch, M. Hamdani, J. Forster, and H. Ney. Analysis of Preprocessing Techniques for Latin Handwriting Recognition. *2012 International Conference on Frontiers in Handwriting Recognition*, pages 280–284, 2012.

[11] R. Smith. An overview of the tesseract OCR engine. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2:629–633, 2007.