Development of training data for Optical Character recognition using deformed printing characters

Ken Kariya^a, Takahiro Fujishima^b, Lifeng Zhang^{a*}

Kyushu Institute of Technology, Kitakyushu, 804-8550 Japan

*Corresponding Author: zhang@ele.kyutech.jp

Abstract

In recent years, with the advanced internet technology, the logistic system has changed a lot, and the international trade has broadly spread. As a result, the information printed on the paper needs to be converted to a digital data at the receiver side. Notably, the use of character recognition in industrial applications is increasing. When converting the printed characters to character codes, manually input them to the computer or use a technology called Optical Character Recognition(OCR) are the usual two ways. In some case, OCR software does an excellent job of this even some misses are converting, but for a legal restrained document, the miss converting may cause huge losses if the number or the product name is wrong, which we want to avoid in this research. Therefore, we propose denoising and binarization system for OCR and create training data using deformed printing characters to improve the accuracy of character recognition rate. The simulation result shows a reasonable character recognition accuracy when using the training data.

Keywords: OCR, deformation, training data.

1. Introduction

Most of the B2C businesses transmit the order information through the internet directly, but for a B2B market, due to the security reason, the order information still be sent by a traditional manner, such as a paper-based business letter, or by facsimile. There are many disadvantages for converting a paper transmitted document to the digital text. E.g., noise can make the text image more difficult to read. The non-uniform color of the background in the printed material can make the binarization difficult. Also, character deformation can also happen. The character deformation causes the difference between the training data

for character recognition included in OCR software and the read printing characters. This difference causes incorrect character recognition. To convert printed characters to character codes, we use Optical Character Recognition (OCR). In recent years various algorithms for OCR has been developed. As a result, character recognition accuracy has dramatically improved. However, there is no OCR software with a character recognition accuracy of 100% even now. Factors that make character recognition difficult are noise and non-uniform color of the background. Most OCR software enroll the denoising and binarization processing before starting the character recognition, but certain types of noise cannot be removed. Therefore, the binarization cannot be done correctly, which can cause accuracy rates low. Besides, printed characters are often distorted due to the data transmission system, such as the families can drop down the resolution of the image and made the small font size character unreadable. These distorted characters might be recognized wrongly. In this paper, we introduce denoising and binarization system for OCR, and OCR training data. The proposed system is built using Non-Local means denoising and adaptive Gaussian thresholding. OCR training data is created using deformed printing characters.

2. Principle

2.1 Non-Local Means Denoising

To remove the noise contained in the input image, we use the technique called non-local means denoising[1]. The method base on a simple principle: replacing the color of a pixel with an average of the colors of similar pixels. However, similar pixels are not necessarily close to each other. Therefore, we should scan a vast portion of the image in search of all pixels that resemble the pixel one wants to denoise. Denoising is then done by computing the average

color of these most similar pixels. The similarity is evaluated by comparing a whole window around each pixel, and not just the color. This filter is called non-local means and expressed as

$$NLu(p) = \frac{1}{c(p)} \int f(d(B(p), B(q))) \quad (1)$$

Where $d(B(p), B(q))$ is an Euclidean distance between image patches centered respectively at p and q , f is a decreasing function and $C(p)$ is the normalizing factor.

The denoising of a color image $u = (u_1, u_2, u_3)$ and a certain pixel p expressed as

$$\hat{u}_i(p) = \frac{1}{C(p)} \sum_{q \in B(p,r)} u_i(q) w(p, q) \quad (2)$$

$$C(p) = \sum_{q \in B(p,r)} w(p, q) \quad (3)$$

Where $i = 1, 2, 3$ and $B(p, r)$ indicates a neighborhood centered at p and with size $(2r+1) * (2r+1)$ pixels. The weight $w(p, q)$ depends on the squared Euclidean distance $d^2 = d^2(B(p, f), B(q, f))$ of the $(2f+1) * (2f+1)$ color patches centered respectively at p and q .

$$d^2(B(p, f), B(q, f)) = \frac{1}{3(2f+1)^2} \sum_{i=1}^3 \sum_{j \in B(0,f)} (u_i(p+j) - u_i(q+j))^2 \quad (4)$$

We use an exponential kernel in order to compute the weights $w(p, q)$

$$w(p, q) = \exp\left(-\frac{\max(d^2 - 2\sigma^2, 0.0)}{h^2}\right) \quad (5)$$

where σ indicates the standard deviation of the noise and h is the filtering parameter set depending on the value of σ . The weight function is set in order to average similar patches up to noise. That is, patches with square distances smaller than $2\sigma^2$ are set to 1, while larger distances decrease rapidly accordingly to the exponential kernel.

The weight of the reference pixel p in the average is set to the maximum of the weights in the neighborhood $B(p, r)$. This setting avoids the excessive weighting of the reference point in the average.

2.2 Adaptive Gaussian Threshold

To improve the accuracy of character recognition, we

binarize the input image. When we create a binarized image from a grayscale image, threshold processing is necessary. But binarization with a single threshold can be suboptimal, particularly if the page backgrounds is of uneven darkness. Thus, we use an adaptive Gaussian threshold with the weighted average value of the close region of the target pixel as threshold. Gaussian function is used for weighting.

$$f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (4)$$

Where σ denotes the standard deviation. Figure 2 and Figure 3 depict threshold processing with a single threshold and Gaussian threshold. Figure 4 shows the image after denoising and binarization.

Figure 1: Input image

Figure 2: Single threshold

INDICATOR 800*900
 INDICATOR 800*1100
 INDICATOR 275 (25)
 INDICATOR 550*1400*25
 INDICATOR 1100*1400*25
 SIGNWEIGHT (HDG)
 SIGNWEIGHT (S GREEN)
 MA STAND-G
 MA STAND-Y
 F-18AEM
 F-18AD
 DESIGN FENCE PANEL (DS) -M
 CLAMP-FENCE BAND
 BEAUTY BASE
 CONE BASE
 ASH.CAN W/CUP
 STEEL PATES-A

Figure 3: Gaussian threshold

INDICATOR 800*900
 INDICATOR 800*1100
 INDICATOR 275 (25)
 INDICATOR 550*1400*25
 INDICATOR 1100*1400*25
 SIGNWEIGHT (HDG)
 SIGNWEIGHT (S GREEN)
 MA STAND-G
 MA STAND-Y
 F-18AEM
 F-18AD
 DESIGN FENCE PANEL (DS) -M
 CLAMP-FENCE BAND
 BEAUTY BASE
 CONE BASE
 ASH.CAN W/CUP
 STEEL PATES-A

Figure 4: Denoising and binarization

2.2 OCR Training Data

Printed characters can be deformed, so it may be difficult for OCR software to recognize characters correctly. To improve recognition accuracy, we create training data for Tesseract OCR. Tesseract is open source software for character recognition developed by Google and Hewlett Packard. Tesseract can recognize more than 100 languages and can be trained to recognize various characters, so we create new training data using deformed printing characters.

First, we select the area that contains the characters.

Figure 5: Select character area

Second, we decide what character is included in the selected area and create box file. Box file consists of the characters, coordinates, heights, and widths. And then, we create training data using box file.

3. Experimental Results and Discussion

In this paper we created OCR training data for Tesseract by using Figure 1 and Figure 4, and we compared the recognition results when using two training data.

Figure 6 shows the results of character recognition without denoising and binarization.

UDICATGR 800*900
 INDICATOR 800*1100
 INDICATOR 275 (28)
 DWDICATOR 550*1400*25
 INDIGGDR 1100*1400*25
 SIGG*WEIGPH-(HDG)
 SIGNWEIGWW (S GREEN)
 MA S*LAUH
 MA STAPU-Y
 P-18AWA4
 F-18AD 4
 DESIGN PENGIG PANEL (DS) -M
 CUMP-FEWWE BAND
 BEAUW-Y BE
 OONE BASE
 ASRLCAN W/CWP
 STHEL PATES-A 7

Figure 6: Result (without denoising and binarization)

INDICATOR 800*900
 INDICATOR 800*1100
 INDICATOR 275 (25)
 DWDICATOR 550*1400*25
 BWDICATOR 1100*1400*25
 SIGDBWBIGHT (HDG)
 SIGRNWBIGHY (S GRHEN)
 MA STAWSD-G
 MA STAYMWD-Y
 F-18AH
 F-18AD -
 DESIGN FHNCE PANEL (DS) -M
 CMAMP-FHIWCE BAND
 BBAUTY BASB
 CONE BASB
 ASHUCAN W/CIIP
 STFBL PATBS-A 9-

Figure 7: Result (denoising and binarization)

Among the 17 character strings included in Figure 6, the only one character string “INDICATOR 800*1100” was correctly recognized without any mistakes. On the other hand, In Figure 7, “INDICATOR 800*900”, “INDICATOR 800*1100”, and “INDICATOR 275(25)” were correctly recognized. I think the reason for this is because denoising using non-local means and binarization using Gaussian threshold make characters in the input image clear, thus OCR software can recognize characters more correctly

4. Conclusions and Future Work

In this paper, we employee the denoising and binarization technique to improve the OCR accuracy. Also, training the data using deformed printing characters. The results show that recognition accuracy improves by using the proposed approach. However, the results are evaluated only in a constrained circumstance and needs more example data to improve the efficiency.

In the future, we are planning to implement a system to correct the miss-recognized character strings. In particular, we think a correction system by employ the feature matching technique to the character image.

References

- (1) Antoni Buades, Bartomeu Coll and Jean-Michel Morel : “Non-Local Means Denoising”, Image Processing On Line, Vol. 1, pp. 208-212, 2011
- (2) G Deng, and L.W. Cahill: “An adaptive Gaussian filter for noise reduction and edge detection”, Nuclear Science Symposium and Medical Imaging Conference,, Vol. 3, pp. 1615-1619, 1993
- (3) Ray Smith and Google Inc. : “An overview of the Tesseract OCR Engine”, ICDAR '07 Proceedings of the Ninth International Conference on Document Analysis and Recognition, Vol.2, pp.629-633, 2007